REMARKS

Claims 1-4 and 16 are in the application.

Claims 1, 4, and 16 were rejected under 35 U.S.C. §103(a) as being unpatentable over Matsumoto (U.S. Patent No. 6,425,522). Claims 2 and 3 were rejected under 35 U.S.C. 103(a) as being unpatentable over Matsumoto in view of Naim (U.S. Patent No. 6,779,115).

Matsumoto has a filing date of July 19, 1999. The present application, on the other hand, has claimed priority under 35 U.S.C. 119 based upon Japanese Application No. 11-055860 filed on March 3, 1999, Japanese Application No. 11-092699 filed on March 31, 1999, Japanese Application No. 11-178188 filed on June 24, 1999, and Japanese Application No. 11-181967 filed on June 28, 1999. Certified translations in English of these priority applications (i.e., 11-055860, 11-092699, 11-178188, and 11-181967) accompany this Amendment. Because claims 1-4 and 16 are supported, for instance, by the disclosure in Japanese Application No. 11-092699, the present application has an effective filing date, which antedates the date of Matsumoto. As a result, it is respectfully submitted that Matsumoto is not an effective prior art reference against the present application.

Naim has a filing date of February 18, 2000. The present application, on the other hand, has claimed priority under 35 U.S.C. 119 based upon Japanese Application No. 11-055860 filed on March 3, 1999, Japanese Application No. 11-092699 filed on March 31, 1999, Japanese Application No. 11-178188 filed on June 24, 1999, Japanese Application No. 11-181967 filed on June 28, 1999, and Japanese Application No. 11-347474 filed on December 7, 1999. Certified translations in English of these priority applications (i.e., 11-055860, 11-092699, 11-178188, 11-181967, and 11-347474) accompany this Amendment. Accordingly, the present application has

an effective filing date, which antedates the date of Naim. As a result, it is respectfully submitted that Naim is not an effective prior art reference against the present application.

Furthermore, although the applicants have asserted that Matsumoto and Naim are not effective references against the present application, such assertion is not a representation concerning distinctions and/or similarities between the present invention and Matsumoto and Naim. Applicants reserve their right to traverse any further rejection that may be presented on a similar document or documents having an earlier effective date(s).

Accordingly, withdrawal of the 103 rejections to claims 1-4 and 16 are respectfully requested.

In view of the foregoing remarks, it is believed that all of the claims in this application are believed to be allowable, and an official notice to that effect is solicited.

Please charge any fees incurred by reason of this response and not paid herewith to Deposit Account No. 50-0320.

Respectfully submitted,
FROMMER LAWRENCE & HAUG LLP

By:

Dennis M. Smid
Reg. No. 34,930
(212) 588-0800

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT(S): Nobuyuki KIHARA and Teppei YOKOTA

SERIAL No.: 09/674,441 Group Art Unit: 2143

FILED: November 1, 2000 Examiner: Kyung H Shin

INVENTION: DATA PROCESSING APPARATUS, DATA PROCESSING METHOD, TERMINAL UNIT, AND TRANSMISSION METHOD OF DATA PROCESSING APPARATUS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

CERTIFIED TRANSLATION

Yuka NAKAMURA residing at c/o SUGIURA PATENT OFFICE, 7th floor, Ikebukuro Park Bldg., 49-7, Minami Ikebukuro 2-chome, Toshima-ku, Tokyo, JAPAN, declares:

(1) that she knows well both the Japanese and English languages;

(2) that she translated Japanese Application No. 11-347474 from Japanese to English;

(3) that the attached English translation is a true and correct translation of the above-identified Japanese Application to the best of her knowledge and belief; and

(4) that all statements made of her own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements are made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 USC 1001, and that such false statements may jeopardize the validity of the application or any patent issuing thereon.

November 17, 2004 Yuka Nakamura

Date Yuka NAKAMURA

[Title of Document] Specification

[Title of the Invention] Non-volatile Memory, Reproducing Apparatus, Reproducing Method

[Scope of Claims for a Patent]

[Claim 1]

A non-volatile memory for segmenting a signal data file into blocks each having a predetermined length and adding an attribute file for managing the single data file to each single data file, the non-volatile memory having:

a data area for a reproduction management file for managing a plurality of files, each of which is composed of the blocks; and

a file management area for file management information for managing the plurality of data files and the reproduction management file.

[Claim 2]

The non-volatile memory as set forth in claim 1,

wherein the file management information is a file allocation table file.

[Claim 3]

The non-volatile memory as set forth in claim 1,

wherein the reproduction management file has a header containing an identification code for

identifying management information.

[Claim 4]

The non-volatile memory as set forth in claim 3,

wherein the identification code is also redundantly recorded at an areas apart from the header in the reproduction management file.

[Claim 5]

The non-volatile memory as set forth in claim 1,

wherein the reproduction management file has a header containing revision information that is changed whenever recorded data is updated.

[Claim 6]

The non-volatile memory as set forth in claim 5,

wherein the revision information is also redundantly recorded at an areas apart from the header in the reproduction management file.

[Claim 7]

The non-volatile memory as set forth in claim 1,

wherein the attribute file added to each single data file contains the number of blocks that compose each single data file.

[Claim 8]

The non-volatile memory as set forth in claim

1,

wherein the attribute file added to each single data file contains a unique value that is cumulated for each single data file.

[Claim 9]

The non-volatile memory as set forth in claim 1,

wherein the attribute file added to each single data file contains the initial value of a unique value that is cumulated for each single data file.

[Claim 10]

The non-volatile memory as set forth in claim 1,

wherein each block of each single data file contains the initial value of a unique value that is cumulated for each single data file.

[Claim 11]

The non-volatile memory as set forth in claim 1,

wherein a serial number is assigned to each block of each single data file.

[Claim 12]

The non-volatile memory as set forth in claim 1,

wherein the reproduction management file contains data representing the reproduction order of a plurality of data files recorded in the data area.

[Claim 13]

A non-volatile memory reproducing apparatus for reproducing data from a non-volatile memory for segmenting a signal data file into blocks each having a predetermined length and adding an attribute file for managing the single data file to each single data file, the non-volatile memory having a data area for a reproduction management file for managing a plurality of files, each of which is composed of the blocks, and a file management area for file management information for managing the plurality of data files and the reproduction management file, the apparatus comprising:

first determining means for determining whether a part or all of the file management area has been destroyed;

searching means for searching each block when a part or all of the file management area has been destroyed as the determined result of said first determining means;

second determining means for determining whether or not a block searched by said searching means is the reproduction management file;

third determining means for determining whether or not a block searched by said searching means is the attribute file; and

recovering means for recovering the destroyed file management information corresponding to the

searched reproduction management file and the attribute file when said first determining means and said second determining means have searched attribute files corresponding to the reproduction management file and each single data file.

[Claim 14]

The non-volatile memory reproducing apparatus as set forth in claim 13,

wherein the file management information is a file allocation table file.

[Claim 15]

The non-volatile memory reproducing apparatus as set forth in claim 13,

wherein the reproduction management file has a header containing an identification code for identifying management information.

[Claim 16]

The non-volatile memory reproducing apparatus as set forth in claim 15,

wherein the identification code is also redundantly recorded at an areas apart from the header in the reproduction management file.

[Claim 17]

The non-volatile memory reproducing apparatus as set forth in claim 13,

wherein the reproduction management file has a header containing revision information that is

changed whenever recorded data is updated.

[Claim 18]

The non-volatile memory reproducing apparatus as set forth in claim 17,

wherein the revision information is also redundantly recorded at an areas apart from the header in the reproduction management file.

[Claim 19]

The non-volatile memory reproducing apparatus as set forth in claim 13,

wherein the attribute file added to each single data file contains the number of blocks that compose each single data file.

[Claim 20]

The non-volatile memory reproducing apparatus as set forth in claim 13,

wherein the attribute file added to each single data file contains a unique value that is cumulated for each single data file.

[Claim 21]

The non-volatile memory reproducing apparatus as set forth in claim 13,

wherein the attribute file added to each single data file contains the initial value of a unique value that is cumulated for each single data file.

[Claim 22]

The non-volatile memory reproducing apparatus

as set forth in claim 13,

wherein each block of each single data file contains the initial value of a unique value that is cumulated for each single data file.

[Claim 23]

The non-volatile memory reproducing apparatus as set forth in claim 13,

wherein a serial number is assigned to each block of each single data file.

[Claim 24]

A non-volatile memory reproducing method for reproducing data from a non-volatile memory for segmenting a signal data file into blocks each having a predetermined length and adding an attribute file for managing the single data file to each single data file, the non-volatile memory having a data area for a reproduction management file for managing a plurality of files, each of which is composed of the blocks, and a file management area for file management information for managing the plurality of data files and the reproduction management file, the method comprising the steps of:

(a) determining whether a part or all of the file management area has been destroyed;

(b) searching each block when a part or all of the file management area has been destroyed as the determined result of step (a);

(c) determining whether or not a block searched at step (b) is the reproduction management file;

(d) determining whether or not a block searched at step (b) is the attribute file; and

(e) recovering the destroyed file management information corresponding to the searched reproduction management file and the attribute file when attribute files have been searched corresponding to the reproduction management file and each single data file at steps (a) and (b).

[Detailed Description of the Invention]

[0001]

[Technical Field to which the Invention belongs]

The present invention relates to a non-volatile memory for managing files recorded on a memory card using a FAT (File Allocation Table). In addition, the present invention relates to a non-volatile memory reproducing apparatus for reproducing data from the non-volatile memory, and a method.

[0002]

[Prior Art]

EEPROM (Electrically Erasable Programmable ROM) that is an electrically rewritable memory requires a large space because each bit is composed of two transistors. Thus, the integration of EEPROM is restricted. To solve this problem, a flash memory that

allows one bit to be accomplished with one transistor using all-bit-erase system has been developed. The flash memory is being expected as a successor of conventional record mediums such as magnetic disks and optical discs.

[0003]

A memory card using a flash memory is also known. The memory card can be freely attached to an apparatus and detached therefrom. A digital audio recording/reproducing apparatus that uses a memory card instead of a conventional CD (Compact Disc: Trademark) or MD (Mini Disc: Trademark) can be accomplished.

[0004]

A file management system used for a conventional personal computer is named FAT (File Allocation Table). In the FAT system, when a particular file is defined, predetermined parameters are successively set to the file. Thus, the size of a file becomes variable. One file is composed of at least one management unit (sector, cluster, or the like). Data corresponding to the management unit is written to a table referred to as FAT. In the FAT file system, a file structure can be easily formed regardless of the physical characteristics of a record medium. Thus, the FAT file system can be used for a magneto-optical disc as well as a floppy disk and a hard disk. In the above-mentioned memory card, the FAT

file system is used.

[0005]

[Problem to be Solved by the Invention]

In the conventional FAT file system, once the FAT is destroyed, it cannot be almost recovered. Thus, as possible countermeasures, it is necessary to backup data to another medium. Among users of personal computers, such countermeasures are essential. Thus, the users should backup data as their responsibilities. However, it is troublesome for the users to backup data. In addition, to do that, another medium is required.

[0006]

Therefore, an object of the present invention is to provide a non-volatile memory, a data apparatus, and a method that allow a file to be recovered even if a file management table is destroyed without need to make a backup file.

[0007]

[Means for Solving the Problem]

A first aspect of the present invention is a non-volatile memory for segmenting a signal data file into blocks each having a predetermined length and adding an attribute file for managing the single data file to each single data file, the non-volatile memory having a data area for a reproduction management file for managing a plurality of files, each of which is

composed of the blocks, and a file management area for file management information for managing the plurality of data files and the reproduction management file.

[0008]

A second aspect of the present invention is a non-volatile memory reproducing apparatus for reproducing data from a non-volatile memory for segmenting a signal data file into blocks each having a predetermined length and adding an attribute file for managing the single data file to each single data file, the non-volatile memory having a data area for a reproduction management file for managing a plurality of files, each of which is composed of the blocks, and a file management area for file management information for managing the plurality of data files and the reproduction management file, the apparatus comprising a first determining means for determining whether a part or all of the file management area has been destroyed, a searching means for searching each block when a part or all of the file management area has been destroyed as the determined result of the first determining means, a second determining means for determining whether or not a block searched by the searching means is the reproduction management file, a third determining means for determining whether or not a block searched by the searching means is the attribute file, and a recovering means for recovering

the destroyed file management information corresponding to the searched reproduction management file and the attribute file when the first determining means and the second determining means have searched attribute files corresponding to the reproduction management file and each single data file.

[0009]

A third aspect of the present invention is a non-volatile memory reproducing method for reproducing data from a non-volatile memory for segmenting a signal data file into blocks each having a predetermined length and adding an attribute file for managing the single data file to each single data file, the non-volatile memory having a data area for a reproduction management file for managing a plurality of files, each of which is composed of the blocks, and a file management area for file management information for managing the plurality of data files and the reproduction management file, the method comprising the steps of (a) determining whether a part or all of the file management area has been destroyed, (b) searching each block when a part or all of the file management area has been destroyed as the determined result of step (a), (c) determining whether or not a block searched at step (b) is the reproduction management file, (d) determining whether or not a block searched at step (b) is the attribute file, and (e) recovering

the destroyed file management information corresponding to the searched reproduction management file and the attribute file when attribute files have been searched corresponding to the reproduction management file and each single data file at steps (a) and (b).

[0010]

According to the present invention, the record media ,such as non-volatile memory that has the data area and the file management area. The attribute file that manage data file is recorded in terms of files on the data area. The reproduction management file that manage a plurality of files is recorded on the data area. The plurality of data file and the reproduction management file are recorded in file management area. When the non-volatile memory is to be reproduced, in the case in which the file management area is destroyed partly or wholly, the file can be recovered corresponding to the reproduction management file and the attribute file.

[0011]

[Embodiment of the Invention]

Next, an embodiment of the present invention will be described. Fig. 1 is a block diagram showing the structure of a digital audio recorder/player using a memory card according to an embodiment of the present invention. The digital audio recorder/player records and reproduces a digital audio signal using a

attachable/detachable memory card. In reality, the recorder/player composes an audio system along with an amplifying unit, a speaker, a CD player, an MD recorder, a tuner, and so forth. However, it should be noted that the present invention can be applied to other audio recorders. In other words, the present invention can be applied to a portable recording/reproducing apparatus. In addition, the present invention can be applied to a set top box that records a digital audio data that is circulated as a satellite data communication, a digital broadcast, or Internet. Moreover, the present invention can be applied to a system that records/reproduces moving picture data and still picture data rather than audio data. The system according to the embodiment of the present invention can record and reproduce additional information such as picture and text other than a digital audio signal.

[0012]

The recording/reproducing apparatus has an audio encoder/decoder IC 10, a security IC 20, a DSP (Digital Signal Processor) 30. Each of these devices is composed of a one-chip IC. The recording/reproducing apparatus has a attachable/detachable memory card 40. The one-chip IC of the memory card 40 has flash memory (non-volatile memory), a memory control block, and a security block.

The security block has a DES (Data Encryption Standard) encrypting circuit. According to the embodiment, the recording/reproducing apparatus may use a microcomputer instead of the DSP 30.

[0013]

The audio encoder/decoder IC 10 has an audio interface 11 and an encoder/decoder block 12. The encoder/decoder block 12 encodes a digital audio data corresponding to a highly efficient encoding method and writes the encoded data to the memory card 40. In addition, the encoder/decoder block 12 decodes encoded data that is read from the memory card 40. As the highly efficient encoding method, the ATRAC3 format that is a modification of the ATRAC (Adaptive Transform Acoustic Coding) format used in Mini-Disc is used.

[0014]

In the ATRAC3 format, audio data sampled at 44.1 kHz and quantized with 16 bits is highly efficiently encoded. In the ATRAC3 format, the minimum data unit of audio data that is processed is a sound unit (SU). 1 SU is data of which data of 1024 samples (1024 x 16 bits x 2 channels) is compressed to data of a several hundreds bytes. The duration of 1 SU is around 23 msec. In the highly efficient encoding method, the data amount of audio data is compressed to data that is around 10 times smaller than that of original data. As with the ATRAC1 format used in Mini-

Disc, the audio signal compressed and decompressed corresponding to the ATRAC3 format less deteriorates in the audio quality.

[0015]

A line input selector 13 selectively supplies the reproduction output signal of an MD, the output signal of a tuner, or a reproduction output signal of a tape to an A/D converter 14. The A/D converter 14 converts the input line signal to a digital audio signal (sampling frequency = 44.1 kHz; the number of quantizing bits = 16). A digital input selector 16 selectively supplies a digital output signal of an MD, a CD, or a CS (Satellite Digital Broadcast) to a digital input receiver 17. The digital input signal is transmitted through for example an optical cable. An output signal of the digital input receiver 17 is supplied to a sampling rate converter 15. The sampling rate converter 15 converts the digital input signal into a digital audio signal (sampling frequency = 44.1 kHz; the number of quantizing bits = 16).

[0016]

The encoder/decoder block 12 of the audio encoder/decoder IC 10 supplies encoded data to a DES encrypting circuit 22 through an interface 21 of the security IC 20. The DES encrypting circuit 22 has a FIFO 23. The DES encrypting circuit 22 is disposed so as to protect the copyright of contents. The memory

card 40 also has a DES encrypting circuit. The DES encrypting circuit 22 of the recording/reproducing apparatus has a plurality of master keys and an apparatus-unique storage key. The DES encrypting circuit 22 also has a random number generating circuit. The DES encrypting circuit 22 can share an authenticating process and a session key with the memory card 40 that has the DES encrypting circuit. In addition, the DES encrypting circuit 22 can re-encrypt data with the storage key of the DES encrypting circuit.

[0017]

The encrypted audio data that is output from the DES encrypting circuit 22 is supplied to a DSP (Digital Signal Processor) 30. The DSP 30 communicates with the memory card 40 through an interface. In this example, the memory card 40 is attached to an attaching/detaching mechanism (not shown) of the recording/reproducing apparatus. The DSP 30 writes the encrypted data to the flash memory of the memory card 40. The encrypted data is serially transmitted between the DSP 30 and the memory card 40. In addition, an external SRAM (Static Random Access Memory) 31 is connected to the DSP 30. The SRAM 31 provides the recording/reproducing apparatus with a sufficient storage capacity so as to control the memory card 40.

[0018]

A bus interface 32 is connected to the DSP 30. Data is supplied from an external controller (not shown) to the DSP 30 through a bus 33. The external controller controls all operations of the audio system. The external controller supplies data such as a record command or a reproduction command that is generated corresponding to a user's operation through an operation portion to the DSP 30 through the bus interface 32. In addition, the external controller supplies additional information such as image information and character information to the DSP 30 through the bus interface 32. The bus 33 is a bidirectional communication path. Additional information that is read from the memory card 40 is supplied to the external controller through the DSP 30, the bus interface 32, and the bus 33. In reality, the external controller is disposed in for example an amplifying unit of the audio system. In addition, the external controller causes a display portion to display additional information, the operation state of the recorder, and so forth. The display portion is shared by the audio system. Since data that is exchanged through the bus 33 is not copyright protected data, it is not encrypted.

[0019]

The encrypted audio data that is read from the memory card 40 by the DSP 30 is decrypted by the

security IC 20. The audio encoder/decoder IC 10 decodes the encoded data corresponding to the ATRAC3 format. Output data of the audio encoder/decoder 10 is supplied to a D/A converter 18. The D/A converter 18 converts the output data of the audio encoder/decoder 10 into an analog signal. The analog audio signal is supplied to a line output terminal 19.

[0020]

The analog audio signal is supplied to an amplifying unit (not shown) through the line output terminal 19. The analog audio signal is reproduced from a speaker or a head set. The external controller supplies a muting signal to the D/A converter 18. When the muting signal represents a mute-on state, the external controller prohibits the audio signal from being output from the line output terminal 19.

[0021]

Fig. 2 is a block diagram showing the internal structure of the DSP 30. Referring to Fig. 2, the DSP 30 comprises a core 34, a flash memory 35, an SRAM 36, a bus interface 37, a memory card interface 38, and inter-bus bridges. The DSP 30 has the same function as a microcomputer. The core 34 is equivalent to a CPU. The flash memory 35 stores a program that causes the DSP 30 to perform predetermined processes. The SRAM 36 and the external SRAM 31 are used as a RAM of the recording/reproducing apparatus.

[0022]

The DSP 30 controls a writing process for writing encrypted audio data and additional information to the memory card 40 corresponding to an operation signal such as a record command received through the bus interfaces 32 and 37 and a reading process for reading them therefrom. In other words, the DSP 30 is disposed between the application software side of the audio system that records/reproduces audio data and additional information and the memory card 40. The DSP 30 is operated when the memory card 40 is accessed. In addition, the DSP 30 is operated corresponding to software such as a file system.

[0023]

The DSP 30 manages files stored in the memory card 40 with the FAT system used in conventional personal computers. In addition to the file system, according to the embodiment of the present invention, a management file is used. The management file will be descried later. The management file is used to manage data files stored in the memory card 40. The management file as the first file management information is used to manage audio data files. On the other hand, the FAT as the second file management information is used to mange all files including audio data files and management files stored in the flash memory of the memory card 40. The management file is

stored in the memory card 40. The FAT is written to the flash memory along with the route directory and so forth before the memory card 40 is shipped. The details of the FAT will be described later.

[0024]

According to the embodiment of the present invention, to protect the copyright of data, audio data that has been compressed corresponding to the ATRAC3 format is encrypted. On the other hand, since it is not necessary to protect the copyright of the management file, it is not encrypted. There are two types of memory cards that are an encryption type and a non-encryption type. However, a memory card for use with the recorder/player that records copyright protected data is limited to the encryption type. Voice data and image data that are recorded by users are recorded on non-encryption type memory cards.

[0025]

Fig. 3 is a block diagram showing the internal structure of the memory card 40. The memory card 40 comprises a control block 41 and a flash memory 42 that are structured as a one-chip IC. A bidirectional serial interface is disposed between the DSP 30 of the recorder/player and the memory card 40. The bidirectional serial interface is composed of ten lines that are a clock line SCK for transmitting a clock signal that is transmitted along with data, a

status line SBS for transmitting a signal that represents a status, a data line DIO for transmitting data, an interrupt line INT, two GND lines, two INT lines, and two reserved lines.

[0026]

The clock line SCK is used for transmitting a clock signal in synchronization with data. The status line SBS is used for transmitting a signal that represents the status of the memory card 40. The data line DIO is used for inputting and outputting a command and encrypted audio data. The interrupt line INT is used for transmitting an interrupt signal that causes the memory card 40 to interrupt the DSP 30 of the recorder/player. When the memory card 40 is attached to the recorder/player, the memory card 40 generates the interrupt signal. However, according to the embodiment of the present invention, since the interrupt signal is transmitted through the data line DIO, the interrupt line INT is grounded.

[0027]

A serial/parallel converting, parallel/serial converting, and interface block (S/P, P/S, I/F block) 43 is an interface disposed between the DSP 30 of the recorder/player and the control block 41 of the memory card 40. The S/P, P/S, and IF block 43 converts serial data received from the DSP 30 of the recorder/player into parallel data and supplies the parallel data to

the control block 41. In addition, the S/P, P/S, and IF block 43 converts parallel data received from the control block 41 into serial data and supplies the serial data to the DSP 30. When the S/P, P/S, and IF block 43 receives a command and data through the data line DIO, the S/P, P/S, and IF block 43 separates them into these that are normally accessed to the flash memory 42 and those that are encrypted.

[0028]

In the format of which data is transmitted through the data line DIO, after a command is transmitted, data is transmitted. The S/P, P/S, and IF block 43 detects the code of a command and determines whether the command and data are those that are normally accessed or those that are encoded. Corresponding to the determined result, the S/P, P/S, and IF block 43 stores a command that is normally accessed to a command register 44 and stores data that is normally accessed to a page buffer 45 and a write register 46. In association with the write register 46, the memory card 40 has an error correction code encoding circuit 47. The error correction code encoding circuit 47 generates a redundant code that is an error correction code for data temporarily stored in the page buffer 45.

[0029]

Output data of the command register 44, the

page buffer 45, the write register 46, and the error correction code encoding circuit 47 is supplied to a flash memory interface and sequencer (hereinafter, referred to as memory I/F and sequencer) 51. The memory IF and sequencer 51 is an interface disposed between the control block 41 and the flash memory 42 and controls data exchanged therebetween. Data is written to the flash memory through the memory IF and sequencer 51.

[0030]

Audio data that has been compressed corresponding to the ATRAC3 format and written to the flash memory (hereinafter, this audio data is referred to as ATRAC3 data) is encrypted by the security IC 20 of the recorder/player and the security block 52 of the memory card 40 so as to protect the copyright of the ATRAC3 data. The security block 52 comprises a buffer memory 53, a DES encrypting circuit 54, and a non-volatile memory 55.

[0031]

The security block 52 of the memory card 40 has a plurality of authentication keys and a unique storage key for each memory card. The non-volatile memory 55 stores a key necessary for encrypting data. The key stored in the non-volatile memory 55 cannot be analyzed. According to the embodiment, for example, a storage key is stored in the non-volatile memory 55.

The security block 52 also has a random number generating circuit. The security block 52 authenticates an applicable recorder/player and shares a session key therewith. In addition, the security block 52 re-encrypts contents with the storage key through the DSE encrypting circuit 54.

[0032]

For example, when the memory card 40 is attached to the recorder/player, they are mutually authenticated. The security IC 20 of the recorder/player and the security block 52 of the memory card 40 mutually authenticate. When the recorder/player has authenticated the attached memory card 40 as an applicable memory card and the memory card 40 has authenticated the recorder/player as an applicable recorder/player, they are mutually authenticated. After the mutual authenticating process has been successfully performed, the recorder/player and the memory card 40 generate respective session keys and share them with each other. Whenever the recorder/player and the memory card 40 authenticate each other, they generate respective session keys.

[0033]

When contents are written to the memory card 40, the recorder/player encrypts a contents key with a session key and supplies the encrypted data to the memory card 40. The memory card 40 decrypts the

contents key with the session key, re-encrypts the contents key with a storage key, and supplies the contents key to the recorder/player. The storage key is a unique key for each memory card 40. When the recorder/player receives the encrypted contents key, the recorder/player performs a formatting process for the encrypted contents key, and writes the encrypted contents key and the encrypted contents to the memory card 40.

[0034]

In the above section, the writing process for the memory card 40 was described. In the following, the reading process for the memory card 40 will be described. Data that is read from the flash memory 42 is supplied to the page buffer 45, the read register 48, and the error correction circuit 49 through the memory IF and the sequencer 51. The error correcting circuit 49 corrects an error of the data stored in the page buffer 45. Output data of the page buffer 45 that has been error-corrected and the output data of the read register 48 are supplied to the S/P, P/S, and IF block 43. The output data of the S/P, P/S, and IF block 43 is supplied to the DSP 30 of the recorder/player through the above-described serial interface.

[0035]

When data is read from the memory card 40,

the contents key encrypted with the storage key and the contents encrypted with the block key are read from the flash memory 42. The security block 52 decrypts the contents key with the storage key. The security block 52 re-encrypts the decrypted content key with the session key and transmits the re-encrypted contents key to the recorder/player. The recorder/player decrypts the contents key with the received session key and generates a block key with the decrypted contents key. The recorder/player successively decrypts the encrypted ATRAC3 data.

[0036]

A config.ROM 50 is a memory that stores partition information, various types of attribute information, and so forth of the memory card 40. The memory card 40 also has an erase protection switch 60. When the switch 60 is in the erase protection position, even if a command that causes the memory card 40 to erase data stored in the flash memory 42 is supplied from the recorder/player side to the memory card 40, the memory card 40 is prohibited from erasing the data stored in the flash memory 42. An OSC cont. 61 is an oscillator that generates a clock signal that is the reference of the timing of the process of the memory card 40.

[0037]

Fig. 4 is a schematic diagram showing the

hierarchy of the processes of the file system of the computer system that uses a memory card as a storage medium. On the hierarchy, the top hierarchical level is an application process layer. The application process layer is followed by a file management process layer, a logical address management layer, a physical address management layer, and a flash memory access layer. In the above-mentioned hierarchical structure, the file management process layer is the FAT file system. Physical addresses are assigned to individual blocks of the flash memory. The relation between the blocks of the flash memory and the physical addresses thereof does not vary. Logical addresses are addresses that are logically handled on the file management process layer.

[0038]

Fig. 5 is a schematic diagram showing the physical structure of data handled in the flash memory 42 of the memory card 40. In the memory 42, a data unit (referred to as segment) is divided into a predetermined number of blocks (fixed length). One block is divided into a predetermined number of pages (fixed length). In the flash memory, data is erased as each block at a time. Data is written to the flash memory 42 or read therefrom as a page at a time. The size of each block is the same. Likewise, the size of each page is the same. One block is composed of page 0

to page m. For example, one block has a storage capacity of for example 8 KB (kilobytes) or 16 KB. One page has a storage capacity of 512 B (bytes). When one block has a storage capacity of 8 KB, the total storage capacity of the flash memory 42 is 4 MB (512 blocks) or 8 MB (1024 blocks). When one block has a storage capacity of 16 KB, the total storage capacity of the flash memory 42 is 16 MB (1024 blocks), 32 MB (2048 blocks), or 64 MB (4096 blocks).

[0039]

One page is composed of a data portion of 512 bytes and a redundant portion of 16 bytes. The first three bytes of the redundant portion is an overwrite portion that is rewritten whenever data is updated. The first three bytes successively contain a block status area, a page status area, and an update status area. The remaining 13 bytes of the redundant portion are fixed data that depends on the contents of the data portion. The 13 bytes contain a management flag area (1 byte), a logical address area (2 bytes), a format reserve area (5 bytes), a dispersion information ECC area (2 bytes), and a data ECC area (3 bytes). The dispersion information ECC area contains redundant data for an error correction process against the management flag area, the logical address area, and the format reserve area. The data ECC area contains redundant data for an error correction process against 512-byte

data.

[0040]

The management flag area contains a system flag (1: user block, 0: boot block), a conversion table flag (1: invalid, 0: table block), a copy prohibition flag (1: OK, 0: NG), and an access permission flag (1: free, 0: read protect).

[0041]

The first two blocks - blocks 0 and 1 are boot blocks. The block 1 is a backup of the block 0. The boot blocks are top blocks that are valid in the memory card. When the memory card is attached to the recorder/player, the boot blocks are accessed at first. The remaining blocks are user blocks. Page 0 of the boot block contains a header area, a system entry area, and a boot and attribute information area. Page 1 of the boot block contains a prohibited block data area. Page 2 of the boot block contains a CIS (Card Information Structure)/IDI (identify Drive Information) area.

[0042]

The header area of the boot block contains a boot block ID and the number of effective entries. The system entries are the start position of prohibited block data, the data size thereof, the data type thereof, the data start position of the CIS/IDI area, the data size thereof, and the data type thereof. The

boot and attribute information contains the memory card type (read only type, rewritable type, or hybrid type), the block size, the number of blocks, the number of total blocks, the security/non-security type, the card fabrication data (date of fabrication), and so forth.

[0043]

Since the flash memory has a restriction for the number of rewrite times due to the deterioration of the insulation film, it is necessary to prevent the same storage area (block) from being concentratedly accessed. Thus, when data at a particular logical address stored at a particular physical address is rewritten, updated data of a particular block is written to a non-used block rather than the original block. Thus, after data is updated, the relation between the logical address and the physical address changes. This process is referred to as swap process. Consequently, the same block is prevented from being concentratedly accessed. Thus, the service life of the flash memory can be prolonged.

[0044]

The logical address associates with data written to the block. Even if the block of the original data is different from the block of updated data, the address on the FAT does not change. Thus, the same data can be properly accessed. However, since the swap process is performed, a conversion table that

correlates logical addresses and physical addresses is required (this table is referred to as logical-physical address conversion table). With reference to the logical-physical address conversion table, a physical address corresponding to a logical address designated on the FAT is obtained. Thus, a block designated with a physical address can be accessed.

[0045]

The DSP 30 stores the logical-physical address conversion table in the SRAM. When the storage capacity of the RAM is small, the logical-physical address conversion table can be stored to the flash memory. The logical-physical address conversion table correlates logical addresses (2 bytes) sorted in the ascending order with physical addresses (2 bytes). Since the maximum storage capacity of the flash memory is 128 MB (8192 blocks), 8192 addresses can be assigned with two bytes. The logical-physical address conversion table is managed for each segment. Thus, the size of the logical-physical address conversion table is proportional to the storage capacity of the flash memory. When the storage capacity of the flash memory is 8 MB (two segments), two pages are used as the logical-physical address conversion table for each of the segments. When the conversion table is stored in the flash memory, a predetermined one bit of the management flag area in the redundant portion in each

page represents whether or not the current block is a block containing the logical-physical address conversion table.

[0046]

The above-described memory card can be used with the FAT file system of a personal computer system as with the disc shaped record medium. The flash memory has an IPL area, a FAT area, and a route directory area (not shown in Fig. 5). The IPL area contains the address of a program to be initially loaded to the memory of the recorder/player. In addition, the IPL area contains various types of memory information. The FAT area contains information with respect to blocks (clusters). The FAT has defined unused blocks, next block number, defective blocks, and last block number. The route directory area contains directory entries that are a file attribute, an update date [day, month, year], file size, and so forth.

[0047]

Next, with reference to Fig. 6, a managing method using the FAT table will be described. Fig. 6 is a schematic diagram showing a memory map. The top area of the memory map is a partition table portion. The partition table portion is followed by a block area, a boot sector, a FAT area, a FAT backup area, a root directory area, a sub directory area, and a data area. On the memory map, logical addresses have been

converted into physical addresses corresponding to the logical-physical address conversion table.

[0048]

The boot sector, the FAT area, the FAT backup area, the root directory area, the sub directory area, and the data area are referred to as FAT partition area.

[0049]

The partition table portion contains the start address and the end address of the FAT partition area. The FAT used for a conventional floppy disk does not have such a partition table. Since the first track has only a partition table, there is a blank area.

[0050]

The boot sector contains the size of the FAT structure (12 bit FAT or 16 bit FAT), the cluster size, and the size of each area. The FAT is used to manage the position of a file recorded in the data area. The FAT copy area is a FAT backup area. The route directory area contains file names, start cluster addresses thereof, and various attributes thereof. The route directory area uses 32 bytes per file.

[0051]

The sub directory area is achieved by a directory attribute file as a directory. In the embodiment shown in Fig. 6, the sub directory area has four files named PBLIST.MSF, CAT.MSF, DOG.MSF, and

MAN.MFA. The sub directory area is used to manage file names and record positions on the FAT. In other words, the slot of the file name CAT.MSF is assigned address "10" on the FAT. The slot of the file name DOG.MSF is assigned address "10" on the FAT.

[0052]

An area after cluster 2 is used as a data area. In this embodiment, audio data that has been compressed corresponding to the ATRAC3 format is recorded. The top slot of the file name MAN.MSA is assigned address "110" on the FAT.

[0053]

According to the embodiment of the present invention, audio data with the file name CAT.MSF is recorded to cluster 5 to 8. Audio data of DOG-1 as the first half of the file with the file name DOG.MSF is recorded to clusters 10 to 12. Audio data DOG-2 as the second half of the file with the file name DOG.MSF is recorded in clusters 100 and 101. Audio data with the file name MAN.MSF is recorded in clusters 110 and 111.

[0054]

In the embodiment of the present invention, an example of which a single file is divided into two portions and dispersedly recorded is described. In the embodiment, an area "Empty" in the data area is a recordable area.

[0055]

An area after cluster 200 is used for managing file names. The file CAT.MSF is recorded to cluster 200. The file DOG.MSF is recorded to cluster 201. The file MAN.MSF is recorded to cluster 202. When the positions of the files are changed, the area after cluster 200 is re-arranged.

[0056]

When the memory card is attached, the beginning and the end of the FAT partition area are recorded with reference to the top partition table portion. After the boot sector portion is reproduced, the root directory area and the sub directory area are reproduced. The slot of the reproduction management information PBLIST.MSF in the sub directory area is detected. Thus, the address of the end portion of the slot of the file PBLIST.MSF is obtained.

[0057]

In the embodiment, since address "200" is recorded at the end of the file PBLIST.MSF, cluster 200 is referenced. The area after cluster 200 is used for managing the reproduction order of files. In the embodiment, the file CAT.MSA is the first program. The file DOG.MSA is the second program. The file MAN.MSA is the third program.

[0058]

After the area after cluster 200 is referenced, slots of the files CAT.MSA, DOG.MSA, and

MAN.MSA are referenced. In Fig. 6, the end of the slot of the file CAT.MSA is assigned address "5". The end of the slot of the file DOG.MSA is assigned address "10". The end of the slot of the file MAN.MSA is assigned address "110".

[0059]

When an entry address is searched on the FAT with address "5", cluster address "6" is obtained. When an entry address is searched on the FAT with address "6", cluster address "7" is obtained. When an entry address is searched on the FAT with address "8", code "FFF" that represents the end is obtained.

[0060]

Thus, the file CAT.MSA uses clusters 5, 6, 7, and 8. With reference to clusters 5, 6, 7, and 8 in the data area, an area of ATRAC3 data with the file name CAT.MSA can be accessed.

[0061]

Next, a method for searching the file DOG.MSF that has been dispersedly recorded will be described. The end of the slot of the file DOG.MSA is assigned address "10". When an entry address on the FAT is searched with address "10", cluster address "11" is obtained. When an entry address on the FAT is searched with address "11" is referenced, cluster address "12" is obtained. When an entry address on the FAT is searched with address "12" is referenced, cluster

address "101" is obtained. When entry address "101" is referenced, code "FFF" that represents the end is obtained.

[0062]

Thus, the file DOG.MSF uses clusters 10, 11, 12, 100, and 101. When clusters 10, 11, and 12 are referenced, the first part of ATRAC3 data of the file DOG.MSF can be accessed. When the clusters 100 and 101 are referenced, the second part of ATRAC3 data of the file DOG.MSF can be accessed.

[0063]

In addition, an entry address is searched on the FAT with address "110" that is recorded on the end of the slot on which the file name "MAN.MSA" is recorded. cluster address "111" entries on entry address "110", an entry address "111" is referenced so as to obtain a code "FFF" that represents the end is recorded.

[0064]

Thus, it is clear that the file MAN.MSA uses clusters 110 and 111, when clusters 110 and 111 is referenced, the area on which ATRAC 3 data "MSN.MSA" is actually recorded can be accessed.

[0065]

As described above, data files dispersed in the flash memory can be linked and sequentially reproduced.

[0066]

According to the embodiment of the present invention, in addition to the file management system defined in the format of the memory card 40, the management file is used for managing tracks and parts of music files. The management file is recorded to a user block of the flash memory 42 of the memory card 40. Thus, as will be described later, even if the FAT of the memory card 40 is destroyed, a file can be recovered.

[0067]

The management file is generated by the DSP 30. When the power of the recorder/player is turned on, the DSP 30 determines whether or not the memory card 40 has been attached to the recorder/player. When the memory card has been attached, the DSP 30 authenticates the memory card 40. When the DSP 30 has successfully authenticated the memory card 40, the DSP 30 reads the boot block of the flash memory 42. Thus, the DSP 30 reads the physical-logical address conversion table and stores the read data to the SRAM. The FAT and the route directory have been written to the flash memory of the memory card 40 before the memory card 40 is shipped. When data is recorded to the memory card 40, the management file is generated.

[0068]

In other words, a record command issued by

the remote controller of the user or the like is supplied to the DSP 30 from the external controller through the bus and the bus interface 32. The encoder/decoder IC 10 compresses the received audio data and supplies the resultant ATRAC3 data to the security IC 20. The security IC 20 encrypts the ATRAC3 data. The encrypted ATRAC3 data is recorded to the flash memory 42 of the memory card 40. Thereafter, the FAT and the management file are updated. Whenever a file is updated (in reality, whenever the recording process of audio data is completed), the FAT and the management file stored in the SRAMs 31 and 36 are rewritten. When the memory card 40 is detached or the power of the recorder/player is turned off, the FAT and the management file that are finally supplied from the SRAMs 31 and 36 are recorded to the flash memory 42. Alternatively, whenever the recording process of audio data is completed, the FAT and the management file written in the flash memory 42 may be rewritten. When audio data is edited, the contents of the management file are updated.

[0069]

In the data structure according to the embodiment, additional information is contained in the management file. The additional information is updated and recorded to the flash memory 42. In another data structure of the management file, an additional

information management file is generated besides the track management file. The additional information is supplied from the external controller to the DSP 30 through the bus and the bus interface 32. The additional information is recorded to the flash memory 42 of the memory card 40. Since the additional information is not supplied to the security IC 20, it is not encrypted. When the memory card 40 is detached from the recorder/player or the power thereof is turned off, the additional information is written from the SRAM of the DSP 30 to the flash memory 42.

[0070]

Fig. 7 is a schematic diagram showing the file structure of the memory card 40. As the file structure, there are a still picture directory, a moving picture directory, a voice directory, a control directory, and a music (HIFI) directory. According to the embodiment, music programs are recorded and reproduced. Next, the music directory will be described. The music directory has two types of files. The first type is a reproduction management file BLIST.MSF (hereinafter, referred to as PBLIST). The other type is an ATRAC3 data file A3Dnnnn.MSA that stores encrypted music data. The music directory can stores up to 400 ATRAC3 data files (namely, 400 music programs). ATRAC3 data files are registered to the reproduction management file and generated by the

recorder/player.

[0071]

Fig. 8 is a schematic diagram showing the structure of the reproduction management file. Fig. 9 is a schematic diagram showing the file structure of one ATRAC3 data file. The reproduction management file is a fixed-length file of 16 KB. An ATRAC3 data file is composed of an attribute header and an encrypted music data area for each music program. The attribute data has a fixed length of 16 KB. The structure of the attribute header is similar to that of the reproduction management file.

[0072]

The reproduction management file shown in Fig. 8 is composed of a header, a memory card name NM-1S (for one byte code), a memory card name NM2-S (for two byte code), a program reproduction sequence table TRKTBL, and memory card additional information INF-S. The attribute header (shown in Fig. 9) at the beginning of the data file is composed of a header, a program name NM1 (for one byte code), a program name NM2 (for two byte code), track information TRKINF (such as track key information), part information PRTINF, and track additional information INF. The header contains information of the number of total parts, the attribute of the name, the size of the additional information, and so forth.

[0073]

The attribute data is followed by ATRAC3 music data. The music data is block-segmented every 16 KB. Each block starts with a header. The header contains an initial value for decrypting encrypted data. Only music data of an ATRAC3 data file is encrypted. Thus, other data such as the reproduction management file, the header, and so forth are not encrypted.

[0074]

Next, with reference to Figs. 10A to 10C, the relation between music programs and ATRAC3 data files will be described. One track is equivalent to one music program. In addition, one music program is composed of one ATRAC3 data (see Fig. 9). The ATRAC3 data file is audio data that has been compressed corresponding to the ATRAC3 format. The ATRAC3 data file is recorded as a cluster at a time to the memory card 40. One cluster has a capacity of 16 KB. A pluralities of files are not contained in one cluster. The minimum data erase unit of the flash memory 42 is one block. In the case of the memory card 40 for music data, a block is a synonym of a cluster. In addition, one cluster is equivalent to one sector.

[0075]

One music program is basically composed of one part. However, when a music program is edited, one

music program may be composed of a plurality of parts. A part is a unit of data that is successively recorded. Normally, one track is composed of one part. The connection of parts of a music program is managed with part information PRTINF in the attribute header of each music program. In other words, the part size is represented with part size PRTSIZE (4 bytes) of the part information PRTINF. The first two bytes of the part size PRTSIZE represents the number of total clusters of the current part. The next two bytes represent the positions of the start sound unit (SU) and the end sound unit (SU) of the beginning and last clusters, respectively. Hereinafter, a sound unit is abbreviated as SU. With such a part notation, when music data is edited, the movement of the music data can be suppressed. When music data is edited for each block, although the movement thereof can be suppressed, the edit unit of a block is much larger than the edit unit of a SU.

[0076]

SU is the minimum unit of a part. In addition, SU is the minimum data unit in the case that audio data is compressed corresponding to the ATRAC3 format. 1 SU is audio data of which data of 1024 samples at 44.1 kHz (1024 x 16 bits x 2 channels) is compressed to data that is around 10 times smaller than that of original data. The duration of 1 SU is around

23 msec. Normally, one part is composed of a several thousands of SU. When one cluster is composed of 42 SU, one cluster allows a sound of one second to be generated. The number of parts composing one track depends on the size of the additional information. Since the number of parts is obtained by subtracting the header, the program name, the additional data, and so forth from one block, when there is no additional information, the maximum number of parts (645 parts) can be used.

[0077]

Fig. 10A is a schematic diagram showing the file structure in the case that two music programs of a CD or the like are successively recorded. The first program (file 1) is composed of for example five clusters. Since one cluster cannot contain two files of the first program and the second program, the file 2 starts from the beginning of the next cluster. Thus, the end of the part 1 corresponding to the file 1 is in the middle of one cluster and the remaining area of the cluster contains no data. Likewise, the second music program (file 2) is composed of one part. In the case of the file 1, the part size is 5. The first cluster starts from the address of 0 SU. The last cluster ends at 4th SU.

[0078]

There are four types of edit processes that

are a divide process, a combine process, an erase process, and a move process. The divide process is performed to divide one track into two portions. When the divide process is performed, the number of total tracks increases by one. In the divide process, one file is divided into two files on the file system. Thus, in this case, the reproduction management file and the FAT are updated. The combine process is performed to combine two tracks into one track. When the combine process is performed, the number of total tracks decreases by one. In the combine process, two files are combined into one file on the file system. Thus, when the combine process is performed, the reproduction management file and the FAT are updated. The erase process is performed to erase a track. The track numbers after the track that has been erased decrease one by one. The move process is performed to change the track sequence. Thus, when the erase process or the move process is performed, the reproduction management file and the FAT are updated.

[0079]

Fig. 10B is a schematic diagram showing the combined result of two programs (file 1 and file 2) shown in Fig. 10A. As a result of the combine process, the combined file is composed of two parts. Fig. 10C is a schematic diagram showing the divided result of which one program (file 1) is divided in the middle of

the cluster 2. By the divide process, the file 1 is composed of clusters 0, 1, and the beginning portion of cluster 2. The file 2 is composed of the end portion of cluster 2 and clusters 3 and 4.

[0080]

As described above, according to the embodiment of the present invention, since the part notation is defined, as the combined result (see Fig. 10B), the start position of the part 1, the end position of the part 1, and the end portion of the part 2 can be defined with SU. Thus, to pack the space due to the combined result, it is not necessary to move the music data of the part 2. In addition, as the divided result (see Fig. 10C), it is not necessary to move data and pack the space at the beginning of the file 2.

[0081]

Fig. 11 is a schematic diagram showing the detailed data structure of the reproduction management file PBLIST. Figs. 12A and 12B show a header portion and the remaining portion of the reproduction management file PBLIST. The size of the reproduction management file is one cluster (one block = 16 KB). The size of the header shown in Fig. 12A is 32 bytes. The rest of the reproduction management file PBLIST shown in Fig. 12B contains a name NM1-S area (256 bytes) (for the memory card), a name NM2-S area (512 bytes), a contents key area, a MAC area, an S-YMDhms

area, a reproduction sequence management table TRKTBL area (800 bytes), a memory card additional information INF-S area (14720 bytes), and a header information redundant area. The start positions of these areas are defined in the reproduction management file.

[0082]

The first 32 bytes of (0x0000) to (0x0010) shown in Fig. 12A are used for the header. In the file, 16-byte areas are referred to as slots. Referring to Fig. 12A, the header are placed in the first and second slots. The header contains the following areas. An area denoted by "Reserved" is an undefined area. Normally, in a reserved area, a null (0x00) is written. However, even if any data is written to a reserved area, the data written in the reserved is ignored. In a future version, some reserved areas may be used. In addition, data is prohibited from being written to a reserved area. When an option area is not used, it is treated as a reserved area.

[0083]

BLKID-TL0 (4 bytes)

Meaning: BLOCKID FILE ID

Function: Identifies the top of the reproduction management file.

Value: Fixed value = "TL = 0" (for example, 0x544C2D30)

Mcode (2 bytes)

Meaning: MAKER CODE

Function: Identifies the maker and model of the recorder/player

Value: High-order 10 bits (Maker code); low-order 6 bits (model code).

REVISION (4 bytes)

Meaning: Number of rewrite times of PBLIST

Function: Increments whenever the reproduction management file is rewritten.

Value: Starts at 0 and increments by 1.

S-YMDhms (4 bytes) (Option)

Meaning: Year, month, day, hour, minute, and second recorded by the recorder/player with a reliable clock.

Function: Identifies the last recorded date and time.

Value: bits 25 to 31: Year 0 to 99 (1980 to 2079)

bits 21 to 24: Month 0 to 12

bits 16 to 20: Day 0 to 31

bits 11 to 15: Hour 0 to 23

bits 05 to 10: Minute 0 to 59

bits 00 to 04: Second 0 to 29 (two bit interval)

[0084]

SY1C+L (2 bytes)

Meaning: Attribute of name (one byte code) of memory card written in NM1-S area.

Function: Represents the character code and the language code as one byte code.

Value: Character code (C): High-order one byte

00: Non-character code, binary number

01: ASCII (American Standard Code for Information Interchange)

02: ASCII+KANA

03: Modified 8859-1

81: MS-JIS

82: KS C 5601-1989

83: GB (Great Britain) 2312-80

90: S-JIS (Japanese Industrial Standards) (for Voice)

[0085]

Language code (L): Low-order one byte Identifies the language based on EBU Tech 3258 standard.

00: Not set

08: German

09: English

0A: Spanish

0F: French

15: Italian

1D: Dutch

65: Korean

69: Japanese

75: Chinese

When data is not recorded, this area is all 0.

[0086]

SN2C+L (2 bytes)

Meaning: Attribute of name of memory card in NM2-S area.

Function: Represents the character code and the language coded as one byte code.

Value: Same as SN1C+L

SINFSIZE (2 bytes)

Meaning: Total size of additional information of memory card in INF-S area.

Function: Represents the data size as an increment of 16 bytes. When data is not recorded, this area is all 0.

Value: Size: 0x0001 to 0x39C (924)

T-TRK (2 bytes)

Meaning: TOTAL TRACK NUMBER

Function: Represents the number of total tracks.

Value: 1 to 0x0190 (Max. 400 tracks)

When data is recorded, this area is all 0.

VerNo (2 bytes)

Meaning: Format version number

Function: Represents the major version number (high order one byte) and the minor version number (low order one byte).

Value: 0x0100 (Ver 1.0)

0x0203 (Ver 2.3)

[0087]

Next, areas (see Fig. 13B) that preceded by the header will be described.

[0088]

NM1-S

Meaning: Name of memory card (as one byte code)

Function: Represents the name of the memory card as one byte code (max. 256). At the end of this area, an end code (0x00) is written. The size is calculated from the end code. When data is not recorded, null (0x00) is recorded from the beginning (0x0020) of this area for at least one byte.

Value: Various character code

NM2-S

Meaning: Name of memory card (as two byte code)

Function: Represents the name of the memory card as two byte code (max. 512). At the end of this area, an end code (0x00) is written.

The size is calculated from the end code. When data is not recorded, null (0x00) is recorded from the beginning (0x0120) of this area for at least two bytes.

Value: Various character code

[0089]

CONTENTS KEY

Meaning: Value for music program. Protected with MG(M) and stored. Same as CONTENTS KEY.

Function: Used as a key necessary for calculating MAC of S-YMDhms.

Value: 0 to 0xFFFFFFFFFFFFFFFF

MAC

Meaning: Forged copyright information check value

Function: Represents the value generated with S-YMDhms and CONTENTS KEY.

Value: 0 to 0xFFFFFFFFFFFFFFFF

[0090]

TRK-nnn

Meaning: SQN (sequence) number of ATRAC3 data file reproduced.

Function: Represents FNo of TRKINF.

Value: 1 to 400 (0x190)

When there is no track, this area is all 0.

INF-S

Meaning: Additional information of memory card (for example, information with respect to photos, songs, guides, etc.)

Function: Represents variable length additional information with a header. A plurality of types of additional information may be used. Each of the types of additional information has an ID and a data size. Each additional information area including a header is composed of at least 16 bytes and a multiple of 4 bytes. For details, see the following section.

Value: Refer to the section of "Data Structure of Additional Information".

S-YMDhms (4 bytes) (Option)

Meaning: Year, month, day, hour, minute, and second recorded by the recorder/player with a reliable clock.

Function: Identifies the last recorded date and time. In this case of EMD, this area is mandatory.

Value: bits 25 to 31: Year 0 to 99 (1980 to 2079)

bits 21 to 24: Month 0 to 12

bits 16 to 24: Day 0 to 31

bits 11 to 15: Hour 0 to 23

bits 05 to 10: Minute 0 to 59

bits 00 to 04: Second 0 to 29 (two second interval)

[0091]

As the last slot of the reproduction management file, the same BLKID-TL0, Mcode, and REVISION as those in the header are written.

[0092]

While data is being recorded to a memory card, it may be mistakenly or accidentally detached or the power of the recorder/player may be turned off. When such an improper operation is performed, a defect should be detected. As described above, the REVISION area is placed at the beginning and end of each block. Whenever data is rewritten, the value of the REVISION area is incremented. If a defect termination takes place in the middle of a block, the value of the REVISION area at the beginning of the block does not match the value of the REVISION area at the end of the block. Thus, such a defect termination can be detected. Since there are two REVISION areas, the abnormal termination can be detected with a high probability. When an abnormal termination is detected, an alarm such as an error message is generated.

[0093]

In addition, since the fixed value BLKID-TL0 is written at the beginning of one block (16 KB), when the FAT is destroyed, the fixed value is used as a

reference for recovering data. In other words, with reference to the fixed value, the type of the file can be determined. Since the fixed value BLKID-TL0 is redundantly written at the header and the end portion of each block, the reliability can be secured. Alternatively, the same reproduction management file can be redundantly recorded.

[0094]

The data amount of an ATRAC3 data file is much larger than that of the track information management file. In addition, as will be described later, a block number BLOCK SERIAL is added to ATRAC3 data file. However, since a plurality of ATRAC3 files are recorded to the memory card, to prevent them from become redundant, both CONNUM0 and BLOCK SERIAL are used. Otherwise, when the FAT is destroyed, it will be difficult to recover the file. In other words, one ATRAC3 data file may be composed of a plurality of blocks that are dispersed. To identify blocks of the same file, CONNUM0 is used. In addition, to identify the order of blocks in the ATRAC3 data file, BLOCK SERIAL is used.

[0095]

Likewise, the maker code (Mcode) is redundantly recorded at the beginning and the end of each block so as to identify the maker and the model in such a case that a file has been improperly recorded in

the state that the FAT has not been destroyed.

[0096]

Fig. 12C is a schematic diagram showing the structure of the additional information data. The additional information is composed of the following header and variable length data. The header has the following areas.

[0097]

INF

Meaning: FIELD ID

Function: Represents the beginning of the additional information (fixed value).

Value: 0x69

ID

Meaning: Additional information key code

Function: Represents the category of the additional information.

Value: 0 to 0xFF

SIZE

Meaning: Size of individual additional information

Function: Represents the size of each type of additional information. Although the data size is not limited, it should be at least 16 bytes and a multiple of 4 bytes. The rest of the data should be filled with null (0x00).

Value: 16 to 14784 (0x39C0)

Mcode

Meaning: MAKER CODE

Function: Identifies the maker and model of the recorder/player.

Value: High-order 10 bits (maker code), low-order 10 bits (machine code).

C+L

Meaning: Attribute of characters in data area starting from byte 12.

Function: Represents the character code and the language code as one byte code.

Value: Same as SNC+L

DATA

Meaning: Individual additional information

Function: Represents each type of additional information with variable length data. Real data always starts from byte 12. The length (size) of the real data should be at least 4 bytes and a multiple of 4 bytes. The rest of the data area should be filled with null (0x00).

Value: Individually defined corresponding to the contents of each type of additional information.

[0098]

Fig. 13 is a table that correlates key code values (0 to 63 of additional information and types

thereof. Key code values (0 to 31) are assigned to music character information. Key code values (32 to 63) are assigned to URLs (Uniform Resource Locator) (web information). The music character information and URL information contain character information of the album title, the artist name, the CM, and so forth as additional information.

[0099]

Fig. 14 is a table that correlates key code values (64 to 127) of additional information and types thereof. Key code values (64 to 95) are assigned to paths/others. Key code values (96 to 127) are assigned to control/numeric data. For example, ID = 98 represents TOC-ID as additional information. TOC-ID represents the first music program number, the last music program number, the current program number, the total performance duration, and the current music program duration corresponding to the TOC information of a CD (Compact Disc).

[0100]

Fig. 15 is a table that correlates key code values (128 to 159) of additional information and types thereof. Key code values (128 to 159) are assigned to synchronous reproduction information. In Fig. 15, EMD stands for electronic music distribution.

[0101]

Next, with reference to Figs. 16A to 16E,

real examples of additional information will be described. As with Fig. 12C, Fig. 16A shows the data structure of the additional information. In Fig. 16B, key code ID = 3 (artist name as additional information). SIZE = 0x1C (28 bytes) representing that the data length of additional information including the header is 28 bytes; C+L representing that character code C = 0x01 (ASCII) and language code L = 0x09 (English). Variable length data after byte 12 represents one byte data "SIMON & GRAFUNKEL" as artist name. Since the data length of the additional information should be a multiple of 4 bytes, the rest is filled with (0x00).

[0102]

In Fig. 16C, key code ID = 97 representing that ISRC (International Standard Recording Code: Copyright code) as additional information. SIZE = 0x14 (20 bytes) representing that the data length of the additional information is 20 bytes. C = 0x00 and L = 0x00 representing that characters and language have not been set. Thus, the data is binary code. The variable length data is eight-byte ISRC code representing copyright information (nation, copyright owner, recorded year, and serial number).

[0103]

In Fig. 16D, key code ID = 97 representing recorded date and time as additional information. SIZE

= 0 x 10 (16 bytes) representing that the data length of the additional information is 16 bytes. C = 0x00 and L = representing that characters and language have not been set. The variable length data is four-byte code (32 bit) representing the recorded date and time (year, month, day, hour, minute, second).

[0104]

In Fig. 16E, key code ID = 107 representing a reproduction log as additional information. SIZE = 0x10 (16 bytes) representing that the data length of the additional information is 16 bytes. C = 0x00 and L = 0x00 representing that characters and language have not been set. The variable length data is a four-byte code representing a reproduction log (year, month, day, hour, minute, second). When the recorder/player has a reproduction log function, it records data of 16 bytes whenever it reproduces music data.

[0105]

Fig. 17 is a schematic diagram showing a data arrangement of ATRAC3 data file A3Dnnnn in the case that 1 SU is N bytes (for example, N = 384 bytes). Fig. 17 shows an attribute header (1 block) of a data file and a music data file (1 block). Fig. 17 shows the first byte (0x0000 to 0x7FF0) of each slot of the two blocks (16 x 2 = 32k bytes). As shown in Fig. 18, the first 32 bytes of the attribute header are used as a header; 256 bytes are used as a music program area

NM1 (256 bytes); and 512 bytes are used as a music program title area NM2 (512 bytes). The header of the attribute header contains the following areas.

[0106]

BLKID-HD0 (4 bytes)

Meaning: BLOCKID FIELD ID

Function: Identifies the top of an ATRA3 data file.

Value: Fixed value = "HD = 0" (For example, 0x48442D30)

Mcode (2 bytes)

Meaning: MAKER CODE

Function: Identifies the maker and model of the recorder/player

Value: High-order 10 bits (maker code); low-order 6 bits (machine code)

BLOCK SERIAL (4 bytes)

Meaning: Track serial number

Function: Starets from 0 and increments by 1. Even if a music program is edited, this value does not vary.

Value: 0 to 0xFFFFFFFF.

[0107]

N1C+L (2 bytes)

Meaning: Represents the attribute of data (NM1) of a track (music program title).

Function: Represent the character code and

language code of NM1 as one byte code.

Value: Same as SN1C+L

N2C+L (2 bytes)

Meaning: Represents the attribute of data (NM2) of a track (music program title).

Function: Represent the character code and language code of NM1 as one byte code.

Value: Same as SN1C+L

INFSIZE (2 bytes)

Meaning: Total size of additional information of current track.

Function: Represents the data size as a multiple of 16 bytes. When data is not recorded, this area should be all 0.

Value: 0x0000 to 0x3C6 (966)

T-PRT (2 bytes)

Meaning: Number of total bytes

Function: Represents the number of parts that composes the current track. Normally, the value of T-PRT is 1.

Value: 1 to 285 (645 dec).

T-SU (4 bytes)

Meaning: Number of total SU.

Function: Represents the total number of SU in one track that is equivalent to the program performance duration.

Value: 0x01 to 0x001FFFFF

INX (2 bytes) (Option)

Meaning: Relative position of INDEX

Function: Used as a pointer that represents the top of a representative portion of a music program. The value of INX is designated with a value of which the number of SU is divided by 4 as the current position of the program. This value of INX is equivalent to 4 times larger than the number of SU (around 93 msec).

Value: 0 to 0xFFFF (max, around 6084 sec)

XT (2 bytes) (Option)

Meaning: Reproduction duration of INDEX

Function: Designates the reproduction duration designated by INX-nnn with a value of which the number of SU is divided by 4. The value of INDEX is equivalent to four times larger than the normal SU (around 93 msec).

Value: 0x0000 (no setting); 0x01 to 0xFFFE (up to 6084 sec); 0xFFFF (up to end of music program)

[0108]

Next, the music program title areas NM1 and NM2 will be described.

[0109]

NM1

Means: Character string of music program title

Function: Represents a music program title as one byte code (up to 256 characters) (variable length). The title area should be completed with an end code (0x00). The size should be calculated from the end code. When data is not recorded, null (0x00) should be recorded from the beginning (0x0020) of the area for at least one byte.

Value: Various character codes

NM2

Means: Character string of music program title

Function: Represents a music program title as two byte code (up to 512 characters) (variable length). The title area should be completed with an end code (0x00). The size should be calculated from the end code. When data is not recorded, null (0x100) should be recorded from the beginning (0x0120) of the area for at least two bytes.

Value: Various character codes

[0110]

Data of 80 bytes starting from the fixed position (0x320) of the attribute header is referred to as track information area TRKINF. This area is mainly

used to totally manage the security information and copy control information. Fig. 19 shows a part of TRKINF. The area TRKINF contains the following areas.

[0111]

CONTENTS KEY (8 bytes)

Meaning: Value for each music program. The value of CONTENTS KEY is protected in the security block of the memory card and then stored.

Function: Used as a key for reproducing a music program. It is used to calculate the value of MAC.

Value: 0 to 0xFFFFFFFFFFFFFFFF

MAC (8 bytes)

Meaning: Forged copyright information check value.

Function: Represents the value generated with a plurality of values of TRKINF including contents cumulation numbers and a secret sequence number.

The secret sequence number is a sequence number recorded in the secret area of the memory card. A non-copyright protection type recorder cannot read data from the secret area of the memory card. On the other hand, a copyright protection type recorder and a computer that operates with a program that can read data from a memory card can access the secret area.

[0112]

A (1 byte)

Meaning: Attribute of part.

Function: Represents the information of such as compression mode of a part.

Value: The details will be described in the following (see Fig. 20).

Next, the value of the area A will be described. In the following description, monaural mode (N = 0 or 1) is defined as a special joint mode of which bit 7 = 1, sub signal = 0, main signal = (L+R). A non-copyright protection type player may ignore information of bits 2 and 1.

[0113]

Bit 0 of the area A represents information of emphasis on/off state. Bit 1 of the area A represents information of reproduction skip or normal reproduction. Bit 2 of the area A represents information of data type such as audio data, FAX data, or the like. Bit 3 of the area A is undefined. By a combination of bits 4, 5, and 6, mode information of ATRAC3 is defined as shown in Fig. 20. In other words, N is a mode value of 3 bits. For five types of modes that are monaural (N = 0 or 1), LP (N = 2), SP (N = 4), EX (N = 5), and HQ (N = 7), record duration (64 MB memory card only), data transmission rate, and the number of SU per block are listed. The number of bytes

of 1 SU depends on each mode. The number of bytes of 1 SU in the monaural mode is 136 bytes. The number of bytes of 1 SU in the LP mode is 192 bytes. The number of bytes of 1 SU in the SP mode is 304 bytes. The number of bytes of 1 SU in the EX mode is 384 bytes. The number of bytes of 1 SU in the HQ mode is 512 bytes. Bit 7 of the area A represents ATRAC3 modes (0: Dual, 1: Joint).

[0114]

For example, an example of which a 64 MB memory card is used in the SP mode will be described. A 64-MB memory card has 3968 blocks. In the SP mode, since 1 SU is 304 bytes, one block has 53 SU. 1 SU is equivalent to (1024/44100) seconds. Thus, one block is (1024/44100) x 53 x (3968 - 10) = 4863 seconds = 81 minutes. The transmission rate is (44100/1024) x 304 x 8 = 104737 bps.

[0115]

LT (1 byte)

Meaning: Reproduction restriction flag (bits 7 and 6) and security partition (bits 5 to 0).

Function: Represents a restriction of the current track.

Value: bit 7: 0 = no restriction, 1 = restriction

bit 6: 0 = not expired, 1 = expired

bits 5 to 0: security partition (reproduction prohibited other than 0)

FNo (2 bytes)

Meaning: File number.

Function: Represents the initially recorded track number that designates the position of the MAC calculation value recorded in the secret area of the memory card.

Value: 1 to 0x190 (400)

MG(D) SERIAL-nnn (16 bytes)

Meaning: Represents the serial number of the security block (security IC 20) of the recorder/player.

Function: Unique value for each recorder/player

Value: 0 to 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

CONNUM (4 bytes)

Meaning: Contents cumulation number

Function: Represents a unique value cumulated for each music program. The value is managed by the security block of the recorder/player. The upper limit of the value is 2^{32} that is 4,200,000,000. Used to identify a recorded program.

Value: 0 to 0xFFFFFFFF

[0116]

YMDhms-S (4 bytes) (Option)

Meaning: Reproduction start date and time of track with reproduction restriction

Function: Represents the date and time at which data reproduction is permitted with EMD.

Value: Same as the notation of date and time of other areas

YMDhms-E (4 bytes) (Option)

Meaning: Reproduction end date and time of track with reproduction restriction

Function: Represents the date and time at which data reproduction is expired with EMD.

Value: Same as the notation of date and time of other areas

MT (1 byte) (Option)

Meaning: Maximum value of number of permitted reproduction times

Function: Represents the maximum number of reproduction times designated by EMD.

Value: 1 to 0xFF. When not used, the value of the area MT is 00.

CT (1 byte) (Option)

Meaning: Number of reproduction times

Function: Represents the number of reproduction times in the number of permitted reproduction times. Whenever data is

reproduced, the value of the area CT is gradually reduced.

Value: 0x00 to 0xFF. When not used, the value of the area CT is 0x00. When bit 7 of the area LT is 1 and the value of the area CT is 00, data is prohibited from being reproduced.

[0117]

CC (1 byte)

Meaning: COPY CONTROL

Function: Controls the copy operation.

Value: bits 6 and 7 represent copy control information. bits 4 and 5 represent copy control information of a high speed digital copy operation. bits 2 and 3 represent a security block authentication level. bits 0 and 1 are undefined.

Example of CC:

(bits 7 and 6)

11: Unlimited copy operation permitted

01: copy prohibited

00: one time copy operation permitted

(bits 3 and 2)

00: analog/digital input recording MG authentication level is 0.

When digital record operation using data from a CD is performed, (bits 7 and 6): 00 and (bits 3 and

2): 00.

CN (1 byte) (Option)

Meaning: Number of permitted copy times in high speed serial copy management system

Function: Extends the copy permission with the number of copy times, not limited to one time copy permission and copy free permission. Valid only in first copy generation. The value of the area CN is gradually reduced whenever the copy operation is performed.

"Value"

00: Copy prohibited

01 to 0xFE: Number of times

0xFF: Unlimited copy times

[0118]

The track information area TRKINF is followed by a 24-byte part management information area (PRTINF) starting from 0x0370. When one track is composed of a plurality of parts, the values of areas PRTINF of the individual parts are successively arranged on the time axis. Fig. 22 shows a part of the area PRTINF. Next, areas in the area PRTINF will be described in the order of the arrangement.

[0119]

PRTSIZE (4 bytes)

Meaning: Part size

Function: Represents the size of a part. Cluster: 2 bytes (highest position), start SU: 1 byte (upper), end SU: 1 byte (lowest position).

Value: cluster: 1 to 0x1F40 (8000)

start SU: 0 to 0xA0 (160)

end SU: 0 to 0xA0 (16) (Note that SU starts from 0.)

PRTKEY (8 bytes)

Meaning: Part encrypting value

Function: Encrypts a part. Initial value = 0. Note that edit rules should be applied.

Value: 0 to 0xFFFFFFFFFFFFFFFF

CONNUM0 (4 bytes)

Meaning: Initially generated contents cumulation number key

Function: Uniquely designates an ID of contents.

Value: Same value as the value of the contents cumulation number initial value key

[0120]

As shown in Fig. 17, the attribute header of an ATRAC3 data file contains additional information INF. The additional information is the same as the additional information INF-S (see Figs. 11 and 12B) of the reproduction management file except that the start position is not fixed. The last byte position (a

multiple of four bytes) at the end of one or a plurality of parts is followed by data of the additional information INF.

[0121]

INF

Meaning: Additional information with respect to track

Function: Represents variable length additional information with a header. A plurality of different types of additional information may be arranged. Each of additional information areas has an ID and a data size. Each additional information area is composed of at least 16 bytes and a multiple of 4 bytes.

Value: Same as additional information INF-S of reproduction management file

[0122]

The above-described attribute header is followed by data of each block of an ATRAC3 data file. As shown in Fig. 23, a header is added for each block. Next, data of each block will be described.

[0123]

BLKID-A3D (4 bytes)

Meaning: BLOCKID FILE ID

Function: Identifies the top of ATRAC3 data.

Value: Fixed value = "A3D" (for example,

0x41334420)

Mcode (2 bytes)

Meaning: MAKER CODE

Function: Identifies the maker and model of the recorder/player

Value: High-order 10 bits (maker code); low-order 6 bits (model code)

CONNUMO (4 bytes)

Meaning: Cumulated number of initially created contents

Function: Designates a unique ID for contents. Even if the contents are edited, the value of the area CONNUMO is not changed.

Value: Same as the contents cumulation number initial key

BLOCK SERIAL (4 bytes)

Meaning: Serial number assigned to each track

Function: Starts from 0 and increments by 1. Even if the contents are edited, the value of the area BLOCK SERIAL is not changed.

Value: 0 to 0xFFFFFFFF

BLOCK-SEED (8 bytes)

Meaning: Key for encrypting one block

Function: The beginning of the block is a random number generated by the security block of the recorder/player. The random number is

followed by a value incremented by 1. When the value of the area BLOCK-SEED is lost, since sound is not generated for around one second equivalent to one block, the same data is written to the header and the end of the block. Even if the contents are edited, the value of the area BLOCK-SEED is not changed.

Value: Initially 8-bit random number

INITIALIZATION VECTOR (8 bytes)

Meaning: Value necessary for encrypting/decrypting ATRAC3 data

Function: Represents an initial value necessary for encrypting and decrypting ATRAC3 data for each block. A block starts from 0. The next block starts from the last encrypted 8-bit value at the last SU. When a block is divided, the last eight bytes just before the start SU is used. Even if the contents are edited, the value of the area INITIALIZATION VECTOR is not changed.

Value: 0 to 0xFFFFFFFFFFFFFFFF

SU-nnn

Meaning: Data of sound unit

Function: Represents data compressed from 1024 samples. The number of bytes of output data depends on the compression mode. Even if the contents are edited, the value of the

area SU-nnn is not changed. For example, in the SP mode, N = 384 bytes.

Value: Data value of ATRAC3

[0124]

In Fig. 17, since N = 384, 42 SU are written to one block. The first two slots (4 bytes) of one block are used as a header. In the last slot (two bytes), the areas BLKID-A3D, Mcode, CONNUM0, and BLOCK SERIAL are redundantly written. Thus, M bytes of the remaining area of one block is (16,384 - 384 x 42 - 16 x 3 = 208) bytes. As described above, the eight-byte area BLOCK SEED is redundantly recorded.

[0125]

When the FAT area is destroyed, all blocks of the flash memory are searched. It is determined whether the value of the area ID BLKID at the beginning of each block is TL0, HD0, or A3D. As shown in Figs. 24A to 24C, at step SP1, it is determined whether or not the value of the area ID BLKID at the beginning of the top block is BLKID-TL0.

[0126]

When the determined result at step SP1 is No, the flow advances to step SP2. At step SP2, the block number is incremented. Thereafter, at step SP3, it is determined whether or not the last block has been searched. When the determined result at step SP3 is No, the flow returns to step SP1.

[0127]

When the determined result at step SP1 is Yes, the flow advances to step SP4. At step SP4, it is determined that the searched block is the reproduction management file PBLIST. Thereafter, the flow advances to step SP5. At step SP5, the number of total tracks T-TRK in the reproduction management file PBLIST is stored as N to the register. For example, when the memory has stored 10 ATRAC3 data files (10 music programs), 10 has been stored in T-TRK.

[0128]

Next, at step SP6 with reference to the value of the number of total tracks T-TRK, TRK-001 to TRK-400 of blocks are successively referenced. In this example, since 10 music programs have been recorded, TRK-001 to TRK-010 of blocks are referenced.

[0129]

Since a file number FNO has been recorded in TRK-XXX (where X = 1 to 400) at step SP7, a table that correlates the track number TRK-XXX and the file number FNO is stored to the memory.

[0130]

Next, at step SP8, N stored in the register is gradually decreased. A loop of steps SP6, SP7, and SP8 is repeated until N becomes 0 at step SP9. When the determined result at step SP9 is Yes, the flow advances to step SP10. At step SP10, the pointer is

reset to the top block. The searching process is repeated from the top block.

[0131]

Thereafter, the flow advances to step SP11. At step SP11, it is determined whether or not the value of the area ID BLKID of the top block is BLKID-HD0. When the determined result at step SP11 is No, the flow advances to step SP12. At step SP12, the block number is incremented. At step SP13, it is determined whether or not the last block has been searched.

[0132]

When the determined result at step SP13 is No, the flow returns to step SP11.

[0133]

The searching process is repeated until the determined result at step SP11 becomes Yes. When the determined result at step SP11 is Yes, the flow advances to step SP14. At step SP14, it is determined that the block is the attribute header (see Fig. 8) (0x0000 to 0x03FFF shown in Fig. 18) at the beginning of the ATRAC3 data file.

[0134]

Next, at step SP15, with reference to the file number FNO, the sequence number BLOCK SERIAL of the same ATRAC data file, and the contents cumulation number key CONNUM0 contained in the attribute header, they are stored to the memory. When 10 ATRAC3 data

files have been recorded, since there are 10 blocks of which the value of the area ID BLKID of the top block is BLKID-TL0, the searching process is continued until 10 blocks are searched.

[0135]

When the determined result at step SP13 is Yes, the flow advances to step SP16. At step SP16, the pointer is reset to the top block. The searching process is repeated from the top block.

[0136]

Thereafter, the flow advances to step SP17. At step SP17, it is determined whether or not the value of the area ID BLKID of the top block is BLKID-A3D. When the determined result at step SP17 is No, the flow advances to step SP18. At step SP18, the block number is incremented. Thereafter, at step SP18', it is determined whether or not the last block has been searched. When the determined result at step SP18' is No, the flow returns to step SP17.

[0137]

When the determined result at step SP17 is Yes, the flow advances to step SP19. At step SP19, it is determined that the block contains ATRAC3 data. Thereafter, the flow advances to step SP20. At step SP20, with reference to the serial number BLOCK SERIAL recorded in the ATRAC3 data block and the contents cumulation number key CONNUM0, they are stored to the

memory.

[0138]

In the same ATRAC3 data file, the common number is assigned as the contents cumulation number key CONNUM0 at step SP21. In other words, when one ATRAC3 data file is composed of 10 blocks, a common number is assigned to all the values of the areas CONNUM0.

[0139]

In addition, when one ATRAC3 data is composed of 10 blocks, serial numbers 1 to 0 are assigned to the values of the areas of a plurality of BLOCK SERIAL of the 10 blocks. Corresponding to the values of the areas CONNUM0 and BLOCK SERIAL, it is determined whether the current block composes the same contents and the reproduction order of the current block in the same contents (namely, the connection sequence).

[0140]

When 10 ATRAC3 data files (namely, 10 music programs) have been recorded and each of the ATRAC3 data files is composed of 10 blocks, there are 100 data blocks. With reference to the values of the areas CONNUM0 and BLOCK SERIAL, the reproduction order of music programs of 100 data blocks and the connection order thereof can be obtained.

[0141]

When the determined result at step SP19 is

Yes, all the blocks have been searched for the reproduction management file, the ATRAC3 data file, and the attribute file. Thus, at step SP21, based on the values of the areas CONNUM0, BLOCK SERIAL, FNO, and TRK-X in the order of block numbers of the blocks stored in the memory, the file connection state is obtained. After the connection state is obtained, the FAT may be generated in a free area of the memory.

[0142]

Next, a management file according to a second embodiment of the present invention will be described. Fig. 25 shows the file structure according to the second embodiment of the present invention. Referring to Fig. 25, a music directory contains a track information management file TRKLIST.MSF (hereinafter, referred to as TRKLIST), a backup track information management file TRKLISTB.MSF (hereinafter, referred to as TRKLISTB), an additional information file INFLIST.MSF (that contains an artist name, an ISRC code, a time stamp, a still picture data, and so forth (this file is referred to as INFIST)), an ATRAC3 data file A3Dnnnn.MSF (hereinafter, referred to as A3nnnn). The file TRKLIST contains two areas NAME1 and NAME2. The area NAME1 is an area that contains the memory card name and the program name (for one byte code corresponding to ASCII/8859-1 character code). The area NAME2 is an area that contains the memory card

name and the program name (for two byte code corresponding to MS-JIS/Hankul/Chinese code).

[0143]

Fig. 26 shows the relation between the track information management file TRKLIST, the areas NAME1 and NAME2, and the ATRAC3 data file A3Dnnnn. The file TRKLIST is a fixed-length file of 64k bytes (= 16 k x 4). An area of 32k bytes of the file is used for managing tracks. The remaining area of 32k bytes is used to contain the areas NAME1 and NAME2. Although the areas NAME1 and NAME2 for program names may be provided as a different file as the track information management file, in a system having a small storage capacity, it is convenient to totally manage the track information management file and program name files.

[0144]

The track information area TRKINF-nnnn and part information area PRTINF-nnnn of the track information management file TRKLIST are used to manage the data file A3Dnnnn and the additional information INFLIST. Only the ATRAC3 data file A3Dnnnn is encrypted. In Fig. 26, the data length in the horizontal direction is 16 bytes (0 to F). A hexadecimal number in the vertical direction represents the value at the beginning of the current line.

[0145]

According to the second embodiment, three

files that are the track management file TRKLIST (including a program title file), the additional information management file INFLIST, and the data file A3Dnnnn are used. According to the first embodiment (see Figs. 7, 8, and 9), two files that are the reproduction management file PBLIST for managing all the memory card and the data file ATRAC3 for storing programs are used.

[0146]

Next, the data structure according to the second embodiment will be described. For simplicity, in the data structure according to the second embodiment, the description of similar portions to those of the first embodiment is omitted.

[0147]

Fig. 27 shows the detailed structure of the track information management file TRKLIST.MSN. In the track information management file TRKLIST, one cluster (block) is composed of 16k bytes. The size and data of the file TRKLISTB.MSF are the same as those of the backup file TRKLISTB. The first 32 bytes of the track information management file are used as a header. As with the header of the reproduction management file PBLIST, the header of the file TRKLIST contains a BLKID-TL0/TL1 (backup file ID) area (4 bytes), an area T-TRK (2 bytes) for the number of total tracks, a maker code area Mcode (2 bytes), an area REVISION (4 bytes)

for the number of TRKLIST rewrite times, and an area S-YMDhms (4 bytes) (option) for update date and time data. The meanings and functions of these data areas are the same as those of the first embodiment. In addition, the file TRKLIST contains the following areas.

[0148]

YMDhms (4 bytes)

Represents the last update date (year, month, day) of the file TRKLIST.

N1 (1 byte) (Option)

Represents the sequential number of the memory card (numerator side). When one memory card is used, the value of the area N1 is 0x01.

N2 (1 byte) (Option)

Represents the sequential number of the memory card (denominator side). When one memory card is used, the value of the area N2 is 0x01.

MSID (2 bytes) (Option)

Represents the ID of a memory card. When a plurality of memory cards is used, the value of the area MSID of each memory card is the same (T.B.D.). (T.B.D. (to be defined) represents that this value may be defined in future).

S-TRK (2 bytes).

Represents a special track (T.B.D.). Normally, the value of the area S-TRK is 0x0000.

PASS (2 bytes) (Option)

Represents a password (T.B.D.).

APP (2 bytes) (Option)

Represents the definition of a reproduction application (T.B.D.) (normally, the value of the area APP is 0x0000).

INF-S (2 bytes) (Option)

Represents the additional information pointer of the entire memory card. When there is no additional information, the value of the area INF-S is 0x00.

[0149]

The last 16 bytes of the file TRKLIST are used for an area BLKID-TL0, an area Mcode, and an area REVISION that are the same as those of the header. The backup file TRKLISTB contains the above-described header. In this case, the header contains an area BLKID-TL1, an area Mcode, and an area REVISION.

[0150]

The header is followed by a track information area TRKINF for information with respect to each track and a part information area PRTINF for information with respect to each part of tracks (music programs). Fig. 27 shows the areas preceded by the area TRKLIST. The lower portion of the area TRKLISTB shows the detailed structure of these areas. In Fig. 27, a hatched area represents an unused area.

[0151]

The track information area TRKINF-nnn and the part information area PRTINF-nnn contain areas of an ATRAC3 data file. In other words, the track information area TRKINF-nnn and the part information area PRTINF-nnn each contain a reproduction restriction flag area LT (1 byte), a contents key area CONTENTS KEY (8 bytes), a recorder/player security block serial number area MG(D) SERIAL (16 bytes), an area XT (2 bytes) (option) for representing a feature portion of a music program, an area INX (2 bytes) (option), an area YMDhms-S (4 bytes) (option), an area YMDhms-E (4 bytes) (option), an area MT (1 byte) (option), an area CT (1 byte) (option), an area CC (1 byte) (option), an area CN (1 byte) (option) (these areas YMDhms-S, YMDhms-E, MT, CT, CC, and CN are used for reproduction restriction information and copy control information), an area A (1 byte) for part attribute, a part size area PRTSIZE (4 bytes), a part key area PRTKEY (8 bytes), and a contents cumulation number area CONNUM (4 bytes). The meanings, functions, and values of these areas are the same as those of the first embodiment. In addition, the track information area TRKINF-nnn and the part information area PRTINF-nnn each contain the following areas.

[0152]

T0 (1 byte)

Fixed value (T0 = 0x74)

INF-nnn (Option) (2 bytes)

Represents the additional information pointer (0 to 409) of each track. 00: music program without additional information.

FNM-nnn (4 bytes)

Represents the file number (0x0000 to 0xFFFF) of an ATRK3 data file.

The number nnnn (in ASCII) of the ATRAC3 data file name (A3Dnnnn) is converted into 0xnnnnn.

APP_CTL (4 bytes) (Option)

Represents an application parameter (T.B.D.) (Normally, the value of the area APP_CTL is 0x0000).

P-nnn (2 bytes)

Represents the number of parts (1 to 2039) that compose a music program. This area corresponds to the above-described area T-PART.

PR (1 byte)

Fixed value (PR = 0 x 50).

[0153]

Next, the areas NAME1 (for one byte code) and NAME2 (for two byte code) for managing names will be described. Fig. 28 shows the detailed structure of the area NAME1 (for one byte code area). Each of the areas NAME1 and NAME2 (that will be described later) is segmented with eight bytes. Thus, their one slot is composed of eight bytes. At 0x8000 that is the beginning of each of these areas, a header is placed.

The header is followed by a pointer and a name. The last slot of the area NAME1 contains the same areas as the header.

[0154]

BLKID-NM1 (4 bytes)

Represents the contents of a block (fixed value) (NM1 = 0x4E4D2D31).

PNM1-nnn (4 bytes) (Option)

Represents the pointer to the area NM1 (for one byte code).

PNM1-S

Represents the pointer to a name representing a memory card.

nnn (= 1 to 408) represents the pointer to a music program title.

The pointer represents the start position (2 bytes) of the block, the character code type (2 bits), and the data size (14 bits).

NM1-nnn (Option)

Represents the memory card name and music program title for one byte code (variable length). An end code (0x00) is written at the end of the area.

[0155]

Fig. 29 shows the detailed data structure of the area NAME2 (for two byte code). At 0x8000 that is the beginning of the area, a header is placed. The header is followed by a pointer and a name. The last

slot of the area NAME2 contains the same areas as the header.

[0156]

BLKID-NM2 (4 bytes)

Represents the contents of a block (fixed value) (NM2 = 0x4E4D2D32).

PNM2-nnn (4 bytes) (Option)

Represents the pointer to the area NM2 (for two byte code).

PNM2-S represents the pointer to the name representing the memory card. nnn (= 1 to 408) represents the pointer to a music program title.

The pointer represents the start position (2 bytes) of the block, the character code type (2 bits), and the data size (14 bits).

NM2-nnn (Option)

Represents the memory card name and music program title for two byte code (variable). An end code (0x0000) is written at the end of the area.

[0157]

Fig. 30 shows the data arrangement (for one block) of the ATRAC3 data file A3Dnnnn in the case that 1 SU is composed of N bytes. In this file, one slot is composed of eight bytes. Fig. 30 shows the values of the top portion (0x0000 to 0x3FF8) of each slot. The first four slots of the file are used for a header. As with the data block preceded by the attribute header of

the data file (see Fig. 17) of the first example, a header is placed. The header contains an area BLKID-A3D (4 bytes), a maker code area Mcode (2 bytes), an area BLOCK SEED (8 bytes) necessary for encrypting process, an area CONNUM0 (4 bytes) for the initial contents cumulation number, a serial number area BLOCK SERIAL (4 bytes) for each track, and an area INITIALIZATION VECTOR (8 bytes) necessary for encrypting/decrypting process. The second last slot of the block redundantly contains an area BLOCK SEED. The last slot contains areas BLKID-A3D and Mcode. As with the first embodiment, the header is followed by the sound unit data SU-nnnn.

[0158]

Fig. 31 shows the detailed data structure of the additional information management file INFLIST that contains additional information. In the second embodiment, at the beginning (0x0000) of the file INFLIST, the following header is placed. The header is followed by the following pointer and areas.

[0159]

BLKID-INF (4 bytes)

Represents the contents of the block (fixed value) (INF = 0x494E464F).

T-DAT (2 blocks)

Represents the number of total data areas (0 to 409).

Mcode (2 bytes)

Represents the maker code of the recorder/player

YMDhms (4 bytes)

Represents the record updated date and time.

INF-nnnn (4 bytes)

Represents the pointer to the area DATA of the additional information (variable length, as 2 bytes (slot) at a time). The start position is represented with the high order 16 bits (0000 to FFFF).

DataSlot-0000 (0x0800)

Represents the offset value from the beginning (as a slot at a time).

The data size is represented with low order 16 bits (0001 to 7FFF). A disable flag is set at the most significant bit. MSB = 0 (Enable), MSB = 1 (Disable)

The data size represents the total data amount of the music program.

(The data starts from the beginning of each slot. (The non-data area of the slot is filled with 00.)

The first INF represents a pointer to additional information of the entire album (normally, INF-409).

[0160]

Fig. 32 shows the structure of additional

information. An 8-byte header is placed at the beginning of one additional information data area. The structure of the additional information is the same as that of the first embodiment (see Fig. 12C). In other words, the additional information contains an area IN (2 bytes) as an ID, an area key code ID (1 byte), an area SIZE (2 bytes) that represents the size of each additional information area, and a maker code area Mcode (2 bytes). In addition, the additional information contains an area SID (1 byte) as a sub ID.

[0161]

According to the second embodiment of the present invention, in addition to the file system defined as a format of the memory card, the track information management file TRKLIST for music data is used. Thus, even if the FAT is destroyed, the file can be recovered. Fig. 33 shows a flow of a file recovering process. To recover the file, a computer that operates with a file recovery program and that can access the memory card and a storing device (hard disk, RAM, or the like) connected to the computer are used. The computer has a function equivalent to the DSP30. Next, a file recovering process using the track management file TRKLIST will be described.

[0162]

All blocks of the flash memory whose FAT has been destroyed are searched for TL-0 as the value

(BLKID) at the top position of each block. In addition, all the blocks are searched for NM-1 as the value (BLKID) at the top position of each block. Thereafter, all the blocks are searched for NM-2 as the value (BLKID) at the top position of each block. All the contents of the four blocks (track information management file) are stored to for example a hard disk by the recovery computer.

[0163]

The number of total tracks is obtained from data after the fourth byte of the track information management file. The 20th byte of the track information area TRKINF-001, the value of the area CONNUM-001 of the first music program, and the value of the next area P-001 are obtained. The number of parts is obtained with the value of the area P-001. The values of the areas PRTSIZE of all parts of the track 1 of the area PRTINF is obtained. The number of total blocks (clusters) n is calculated and obtained.

[0164]

After the track information management file is obtained, the flow advances to step 102. At step 102, a voice data file (ATRAC3 data file) is searched. All blocks of other than the management file is searched from the flash memory. Blocks whose top value (BLKID) is A3D are collected.

[0165]

A block of which the value of the area CONNUM0 at the 16th byte of A3Dnnnn is the same as that of the area CONNUM-001 of the first music program of the track information management file and of which the value of the area BLOCK SERIAL that starts from 20th byte is 0 is searched. After the first block is obtained, a block (cluster) with the same value of the area CONNUM value as the first block and of which the value of BLOCK SERIAL is incremented by 1 (1 = 0 + 1) is searched. After the second block is obtained, a block with the same value of the area CONNUM0 as the second block and of which the value of the area BLOCK SERIAL is incremented by 1 (2 = 1 + 1) is searched.

[0166]

By repeating the process, the ATRC3 data file is searched until n blocks (clusters) of the track 1 are obtained. When all the blocks (clusters) are obtained, they are successively stored to the hard disk.

[0167]

The same process for the track 1 is performed for the track 2. In other words, a block of which the value of the area CONNUM0 is the same as that of the area CONNUM-002 of the first music program of the track information management file and of which the value of the area BLOCK SERIAL that starts at the 20th byte is searched. Thereafter, in the same manner as the track

1, the ATRAC3 data file is searched until the last block (cluster) n' is detected. After all blocks (clusters) are obtained, they are successively stored to the hard disk.

[0168]

By repeating the above-described process for all tracks (the number of tracks: m), all the ATRAC3 data is stored to the hard disk controlled by the recovering computer.

[0169]

At step 103, the memory card whose the FAT has been destroyed is re-initialized and then the FAT is reconstructed. A predetermined directory is formed in the memory card. Thereafter, the track information management file and the ATRAC3 data file for m tracks are copied from the hard disk to the memory card. Thus, the recovery process is finished.

[0170]

In the management file and data file, important parameters (in particular, codes in headers) may be recorded triply rather than doubly. When data is redundantly recorded, the same data may be recorded at any positions as long as they are apart from each other for one page or more.

[0171]

[Effects of the Invention]

According to the present invention, in

addition to file management information defined in the non-volatile memory such as the FAT, another (second) file management information is generated and stored in the non-volatile memory. In the second file management information, a fixed length identification code that represents file management information is added. Thus, even if the FAT is destroyed, a file can be easily recovered with the file management information. According to the present invention, since the file management information has a fixed length identification value, the efficiency of the recovering process can be improved. Thus, it is not necessary for the user to make a backup file.

[0172]

Moreover, in the file management information, important parameters are redundantly recorded. Thus, important parameters can be securely protected. In addition, since information that represents the number of rewrite times of a file is recorded at a position apart from the other by 1 page unit or more, a trouble in the middle of the rewriting process of a file can be detected. Moreover, when a trouble takes place, the cause of the trouble can be easily obtained.

[0173]

According to the present invention, in addition to the concept of the file, part management information is stored. Even if one track (music

program) is composed of a plurality of parts, they can be easily managed. Moreover, since the part management information for parts that compose a track is handled along with the track management information (TRKINF) for tracks, the process can be more easily performed than that of Mini-Disc using links (Link-P).

[0174]

Although the present invention has been shown and described with respect to a best mode embodiment thereof, it should be understood by those skilled in the art that the foregoing and various other changes, omissions, and additions in the form and detail thereof may be made therein without departing from the spirit and scope of the present invention.

[Brief Description of the Drawings]

[Fig. 1]

Block diagram showing the structure of a digital audio recorder/player using a non-volatile memory card according to the present invention.

[Fig. 2]

Block diagram showing the internal structure of a DSP 30 according to the present invention.

[Fig. 3]

Block diagram showing the internal structure of a memory card 40 according to the present invention.

[Fig. 4]

Schematic diagram showing a file management

structure of a memory card as a storage medium according to the present invention.

[Fig. 5]

Schematic diagram showing the physical structure of data in a flash memory 42 of the memory card 40 according to the present invention.

[Fig. 6]

Data structure of the memory card 40 according to the present invention.

[Fig. 7]

Schematic diagram showing the hierarchy of the file structure in the memory card 40.

[Fig. 8]

Schematic diagram showing the data structure of a reproduction management file PBLIST.MSF that is a sub directory stored in the memory card 40.

[Fig. 9]

Schematic diagram showing the data structure in the case that one ATRAC3 data file is divided into blocks with a predetermined unit length and that attribute files are added thereto.

[Fig. 10]

Schematic diagram for explaining the combining process and the dividing process.

[Fig. 11]

Schematic diagram showing the data structure of a reproduction management file PBLIST.

[Fig. 12]

Schematic diagram showing the data structure of the reproduction management file PBLIST.

[Fig. 13]

Table that correlates showing types of additional information data and code values thereof.

[Fig. 14]

Table that correlates showing types of additional information data and code values thereof.

[Fig. 15]

Table that correlates showing types of additional information data and code values thereof;

[Fig. 16]

Schematic diagram showing the data structure of additional information data.

[Fig. 17]

Schematic diagram showing a detailed data structure of an ATRAC3 data file.

[Fig. 18]

Schematic diagram showing the data structure of an upper portion of an attribute header that composes an ATRAC3 data file.

[Fig. 19]

Schematic diagram showing the data structure of a middle portion of the attribute header that composes an ATRAC3 data file.

[Fig. 20]

Table that correlates record modes, record time, and so forth.

[Fig. 21]

Table showing copy control states.

[Fig. 22]

Schematic diagram showing the data structure of a lower portion of the attribute header that composes an ATRAC3 data file.

[Fig. 23]

Schematic diagram showing the data structure of a header of a data block of an ATRAC3 data file.

[Figs. 24]

Flow charts showing a recovering method according to the present invention in the case that an FTA area was destroyed.

[Fig. 25]

Schematic diagram showing the file structure in the memory card 40 according to a second embodiment of the present invention;

[Fig. 26]

Schematic diagram showing the relation between a track information management file TRKLIST.MSF and an ATRAC3 data file A3Dnnnnn.MSA.

[Fig. 27]

Schematic diagram showing the detailed data structure of the track information management file TRKLIST.MSF.

[Fig. 28]

Schematic diagram showing the detailed data structure of NAME1 for managing a name.

[Fig. 29]

Schematic diagram showing the detailed data structure of NAME2 for managing a name.

[Fig. 30]

Schematic diagram showing the detailed data structure of an ATRAC3 data file A3Dnnnnn.MSA.

[Fig. 31]

Schematic diagram showing the detailed data structure of INFLIST.MSF that represents additional information.

[Fig. 32]

Schematic diagram showing the detailed data structure of INFLIST.MSF that represents additional information data.

[Fig. 33]

Flow chart showing a recovering method according to the second embodiment of the present invention in the case that an FTA area was destroyed.

[Description of Reference Numerals]

10 ... Audio encoder/decoder IC, 20 ... Security ID, 30 ... DSP, 40 ... Memory card, 42 ... Flash memory, 52 ... Security block, PBLIST ... Reproduction management file, TRKLIST ... Track information management file, INFLIST.MSF ... Additional information management file,

A3Dnnn.MSA ... Audio data file

[Title of Document] Abstract

[Abstract]

[Subject]

In a case of writing or reading an audio and other data in the flash memory of attachable/detachable memory card, the file can be recovered whenever the file management system FAT is destroyed.

[Solving means]

When each file of a memory card for managing recorded files by using FAT (File Allocation Table) is blocked and a file number and block number are added to each segmented block, the connection state of each block can be reconstructed whenever the FAT is destroyed by attaching an attribute file to each blocked file.

[Selected Drawing] Fig. 12

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT(S): Nobuyuki KIHARA and Teppei YOKOTA

SERIAL No.: 09/674,441 Group Art Unit: 2143

FILED: November 1, 2000 Examiner: Kyung H Shin

INVENTION: DATA PROCESSING APPARATUS, DATA PROCESSING METHOD, TERMINAL UNIT, AND TRANSMISSION METHOD OF DATA PROCESSING APPARATUS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

CERTIFIED TRANSLATION

Yuka NAKAMURA residing at c/o SUGIURA PATENT OFFICE, 7th floor, Ikebukuro Park Bldg., 49-7, Minami Ikebukuro 2-chome, Toshima-ku, Tokyo, JAPAN, declares:

(1) that she knows well both the Japanese and English languages;

(2) that she translated Japanese Application No. 11-181967 from Japanese to English;

(3) that the attached English translation is a true and correct translation of the above-identified Japanese Application to the best of her knowledge and belief; and

(4) that all statements made of her own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements are made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 USC 1001, and that such false statements may jeopardize the validity of the application or any patent issuing thereon.

November 17, 2004 Yuka Nakamura

Date Yuka NAKAMURA

[Title of Document] Specification

[Title of the Invention] Data Processing Apparatus and Method

[Scope of Claims for a Patent]

[Claim 1]

A data processing apparatus for moving contents from an attachable/detachable non-volatile storing medium to a storing device,

wherein contents stored in the storing medium are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the storing medium,

wherein when the contents are moved to the storing device, the first key is decrypted by the second key and the decrypted first key is encrypted by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

wherein the first key encrypted by the third key and the encrypted contents are sent to the storing medium.

[Claim 2]

The data processing apparatus as set forth in claim 1,

wherein the storing device that has received the first key encrypted by the third key and the encrypted contents decrypts the first key by the third

key,

encrypts the decrypted first key by a fourth key unique to the storing device, and

stores the first key encrypted by the fourth key.

[Claim 3]

The data processing apparatus as set forth in claim 1,

wherein the contents are managed in the unit of a file, the first key being created for each file, the first key encrypted by the third key being sent as file management information.

[Claim 4]

The data processing apparatus as set forth in claim 3,

wherein the file management information contains information for identifying a file.

[Claim 5]

A data processing apparatus for moving contents from a storing device to an attachable/detachable non-volatile storing medium,

wherein contents stored in the storing device are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the storing device,

wherein when the contents are moved to the storing medium, the first key is decrypted by the

second key and the decrypted first key is encrypted by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

wherein the first key encrypted by the third key and the encrypted contents are sent to the storing medium.

[Claim 6]

The data processing apparatus as set forth in claim 5,

wherein the storing medium that has stored the first key encrypted by the third key and the encrypted contents decrypts the first key by the third key,

encrypts the decrypted first key by a fourth key unique to the storing medium, and

stores the first key encrypted by the fourth key.

[Claim 7]

The data processing apparatus as set forth in claim 5,

wherein the contents are managed in the unit of a file, the first key being created for each file, the first key encrypted by the third key being sent as file management information.

[Claim 8]

The data processing apparatus as set forth in

claim 7,

wherein the file management information contains information for identifying a file.

[Claim 9]

The data processing apparatus as set forth in claim 5,

wherein move history information of the contents is stored to a non-volatile memory different from the storing medium that stores the encrypted contents.

[Claim 10]

The data processing apparatus as set forth in claim 9,

wherein when the contents are moved, if the move history information of the contents is present, with reference thereto, the contents are prohibited from being moved.

[Claim 11]

The data processing apparatus as set forth in claim 5,

wherein the storing device further encrypts contents that are input from the outside, encrypts a first key for encrypting the contents by a second key unique to the storing device, and stores the encrypted contents and the encrypted key.

[Claim 12]

The data processing apparatus as set forth in

claim 5,

wherein the storing device further interfaces with a terminal unit of an electronic contents delivering system.

[Claim 13]

A data processing method for moving contents from an attachable/detachable non-volatile storing medium to a storing device,

wherein contents stored in the storing medium are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the storing medium, the method comprising the steps of:

when the contents are moved to the storing device, decrypting the first key by the second key and encrypting the decrypted first key by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

sending the first key encrypted by the third key and the encrypted contents to the storing medium.

[Claim 14]

A data processing method for moving contents from a storing device to an attachable/detachable non-volatile storing medium,

wherein contents stored in the storing device are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the

storing device, the method comprising the steps of:

when the contents are moved to the storing medium, decrypting the first key by the second key and encrypting the decrypted first key by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

sending the first key encrypted by the third key and the encrypted contents to the storing medium.

[Claim 15]

The data processing method as set forth in claim 14,

wherein move history information of the contents is stored to a non-volatile memory different from the storing medium that stores the encrypted contents.

[Claim 16]

The data processing method as set forth in claim 14,

wherein when the contents are moved, if the move history information of the contents is present, with reference thereto, the contents are prohibited from being moved.

[Detailed Description of the Invention]

[0001]

[Technical Field to which the Invention belongs]

The present invention relates to a data

processing method and a data processing method for applying a record/reproduction of an audio data by using an attachable/detachable memory card for example, devices.

[0002]

[Prior Art]

EEPROM (Electrically Erasable Programmable ROM) that is an electrically rewritable non-volatile memory requires a large space because each bit is composed of two transistors. Thus, the integration of EEPROM is restricted. To solve this problem, a flash memory that allows one bit to be accomplished with one transistor using all-bit-erase system has been developed. The flash memory is being expected as a successor of conventional record mediums such as magnetic disks and optical discs.

[0003]

A memory card using a flash memory is also known. The memory card can be freely attached to an apparatus and detached therefrom. A digital audio recording/reproducing apparatus that uses a memory card instead of a conventional CD (Compact Disc) or MD (Mini Disc) can be accomplished.

[0004]

Since an audio recorder using a memory card as a record medium records and reproduces digital data, when the audio recorder uses a compressing method that

allows data to be reproduced with relatively high quality, the copyright of music data and so forth that are recorded and reproduced should be protected. As an example, using encrypting technologies, memory cards that are not authenticated can be prohibited from being used. In other words, in a combination of an authenticated recorder and an authenticated memory card, encrypted data is decrypted.

[0005]

Conventional memory cards do not have an encrypting function. Thus, to record confidential data to a memory card, the recorder should encrypt data and record the encrypted data to a memory card. However, when a decrypting key is stored in a memory card, the security of data cannot be protected. On the other hand, when a decrypting key is stored in the recorder, encrypted data cannot be decrypted by other than the recorder that has encrypted the data. Thus, the compatibility of the memory cards cannot be maintained. For example, data stored in a memory card of one user cannot be decrypted by a recorder of another user. To solve this problem, a system of which both a recorder and a memory card have respective encrypting functions and they authenticate each other for assuring the security of data and the compatibility of the cards has been proposed.

[0006]

On the other hand, as digital audio/video information and multimedia systems are becoming common, a music data delivering service of which music data is delivered from a music data delivering server to a personal computer through a network such as Internet or a digital broadcast is being accomplished. In such a service, delivered contents data is stored in a hard disk of the personal computer.

[0007]

In a system of which a hard disk is used as an audio server, audio contents data is moved from the hard disk to a memory card. With the memory card, the moved data can be reproduced by for example a portable player. In contrast, audio data is moved from the memory card to the hard disk of the personal computer. In this case, data is moved from the hard disk to a memory card so that the data is not left in the hard disk.

[0008]

In the system of which a hard disk is used as an audio server, when data is moved from a memory card to the hard disk, all contents data of the memory card is moved to the hard disk. In this method, since an encrypting process or the like is not required, the structure becomes simple and the data can be moved at high speed. In addition, since the hard disk cannot decrypt stored data, from a viewpoint of the copyright

owner, this method is the most safety method.

[0009]

[Problem to be solved by the Invention]

However, in the case that a contents key stored in the memory card is encrypted with a storage key of the memory card, when the contents key stored in the hard disk is returned to the memory card, other than the original memory card cannot decrypt encrypted data. In other words, even if contents data is moved from the hard disk to another memory card, the contents data cannot be reproduced. In addition, when the original memory card is lost or destroyed, all data stored therein cannot be used.

[0010]

An object of the present invention is to solve a problem with respect to a data moving operation and provide a data processing apparatus and method that allow a storage key to be used for a storing unit such as a hard disk and to be re-keyed.

[0011]

Another object of the present invention is to provide a data processing apparatus and method for substantially preventing all contents data of a hard disk from being copied to many memory cards.

[0012]

[Means for Solving the Problem]

To solve the above-mentioned problem,

according to the present invention of claim 1, there is provided a data processing apparatus for moving contents from an attachable/detachable non-volatile storing medium to a storing device,

wherein contents stored in the storing medium are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the storing medium,

wherein when the contents are moved to the storing device, the first key is decrypted by the second key and the decrypted first key is encrypted by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

wherein the first key encrypted by the third key and the encrypted contents are sent to the storing medium.

[0013]

According to the present invention of claim 5, there is provided a data processing apparatus for moving contents from a storing device to an attachable/detachable non-volatile storing medium,

wherein contents stored in the storing device are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the storing device,

wherein when the contents are moved to the

storing medium, the first key is decrypted by the second key and the decrypted first key is encrypted by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

wherein the first key encrypted by the third key and the encrypted contents are sent to the storing medium.

[0014]

To securely prevent contents from being illegally copied, storing move history information of the contents in a non-volatile memory is recommended rather than the record medium for storing encrypted contents. In addition, when the move of contents is processed, the move of contents is prohibited by referring move history information of the contents, if move history information of the contents resides in a non-volatile memory.

[0015]

According to the present invention of claim 13, there is provided a data processing method for moving contents from an attachable/detachable non-volatile storing medium to a storing device,

wherein contents stored in the storing medium are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the storing medium, the method comprising the steps of:

when the contents are moved to the storing device, decrypting the first key by the second key and encrypting the decrypted first key by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

sending the first key encrypted by the third key and the encrypted contents to the storing medium.

[0016]

According to the present invention of claim 14, there is provided a data processing method for moving contents from a storing device to an attachable/detachable non-volatile storing medium,

wherein contents stored in the storing device are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the storing device, the method comprising the steps of:

when the contents are moved to the storing medium, decrypting the first key by the second key and encrypting the decrypted first key by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

sending the first key encrypted by the third key and the encrypted contents to the storing medium.

[0017]

This invention is to provide a storing apparatus of a hard disk or the like having a unique key for the apparatus. When an encrypted contents at the first key are moved from the record medium such as the memory card, the first key for encrypting the contents re-encrypts by a unique key of the apparatus with changing the first key into a unique key for the record medium. That is the process of a key lock is repeated. Accordingly the repetition, whenever the contents are moved to the other record medium from the original record medium in a storing apparatus that can be encrypted.

[0018]

Besides, according to an embodiment of the present invention, move history information of the contents for the record medium is stored in the other non-volatile memory instead of a non-volatile memory of a storing apparatus that can be securely prevented the contents from being illegally copied. In other words, if move history information of the contents for the record medium isn't stored in a non-volatile memory, a copied hard disk group is prepared by an execution of a physical entire copy for a storing apparatus such as a hard disk. Secondly, by the replacement of a moved hard disk to a copied hard disk group before the movement, it is allowing to copy a real content toward a majority of memory card. In the process of the move,

the re-movement of a moved content is prohibited by referring move history information of the contents that prevents securely the contents from being illegally copied.

[0019]

[Embodiment of the Invention]

Next, an embodiment of the present invention will be described. Fig. 1 is a block diagram showing the structure of a digital audio signal recorder (recorder/player) using a memory card according to an embodiment of the present invention. The digital audio recorder records and reproduces a digital audio signal using a detachable memory card. For example, the recorder composes an audio system along with an amplifying unit, a speaker, a CD player, an MD recorder, a tuner, and so forth. However, it should be noted that the present invention can be applied to other audio recorders. In other words, the present invention can be applied to a portable recorder. In addition, the present invention can be applied to a recorder that records a digital audio data that is circulated as a satellite data communication, a digital broadcast, or Internet. Moreover, the present invention can be applied to a system that records/reproduces moving picture data and still picture data rather than audio data. The system according to the embodiment of the present invention

can record and reproduce additional information such as picture and text other than a digital audio signal.

[0020]

The recorder has an audio encoder/decoder IC 10, a security IC 20, a DSP (Digital Signal Processor) 30. Each of these devices is composed of a one-chip IC. A detachable memory card 40 is an attachable/detachable memory card toward a recorder. The one-chip IC of the memory card 40 has flash memory (nonvolatile memory), a memory control block, and a security block. The security block has a DES (Data Encryption Standard) encrypting circuit. According to the embodiment, the recording/reproducing apparatus may use a microcomputer instead of the DSP 30.

[0021]

The audio encoder/decoder IC 10 has an audio interface 11 and an encoder/decoder block 12. The encoder/decoder block 12 encodes a digital audio data corresponding to a highly efficient encoding method and writes the encoded data to the memory card 40. In addition, the encoder/decoder block 12 decodes encoded data that is read from the memory card 40. As the highly efficient encoding method, the format (it is referred to as ATRAC3) that is a modification of the ATRAC (Adaptive Transform Acoustic Coding) format used in Mini-Disc can be used.

[0022]

In the ATRAC3 format, audio data sampled at 44.1 kHz and quantized with 16 bits is highly efficiently encoded. In the ATRAC3 format, the minimum data unit of audio data that is processed is a sound unit (SU). 1 SU is data of which data of 1024 samples (1024 x 16 bits x 2 channels) is compressed to data of several hundred bytes. The duration of 1 SU is around 23 msec. In the highly efficient encoding method, the data amount of audio data is compressed to data that is around 10 times smaller than that of original data. As with the ATRAC1 format used in Mini-Disc, the audio signal is processed by the ATRAC3 format that is less deteriorating in the audio quality by the decompression and decompression corresponding to the ATRAC3 format.

[0023]

A line input selector 13 selectively supplies the reproduction output signal of an MD, the output signal of a tuner, or a reproduction output signal of a tape to an A/D converter 14. The A/D converter 14 converts the input line signal to a digital audio signal (sampling frequency = 44.1 kHz; the number of quantizing bits = 16). A digital input selector 16 selectively supplies a digital output signal of an MD, a CD, or a CS (Satellite Digital Broadcast) to a digital input receiver 17. The digital input signal is transmitted through for example an optical cable. An output signal of the digital input receiver 17 is

supplied to a sampling rate converter 15. The sampling rate converter 15 converts the digital input signal into a digital audio signal (sampling frequency = 44.1 kHz; the number of quantizing bits = 16).

[0024]

The encoder/decoder block 12 of the audio encoder/decoder IC 10 supplies encoded data to a DES encrypting circuit 22 through an interface 21 of the security IC 20. The DES encrypting circuit 22 has a FIFO 23. The DES encrypting circuit 22 is disposed so as to protect the copyright of contents. The memory card 40 also has a DES encrypting circuit. The DES encrypting circuit 22 of the recording/reproducing apparatus has a plurality of master keys and an apparatus-unique storage key. The DES encrypting circuit 22 also has a random number generating circuit. The DES encrypting circuit 22 can share an authenticating process and a session key with the memory card 40 that has the DES encrypting circuit. In addition, the DES encrypting circuit 22 can re-encrypt data with the storage key of the DES encrypting circuit.

[0025]

The encrypted audio data that is output from the DES encrypting circuit 22 is supplied to a DSP (Digital Signal Processor) 30. The DSP 30 communicates with the memory card 40 which is attached to an

attaching/detaching mechanism (not shown) of the data processing apparatus via an interface. The DSP 30 writes the encrypted data to the flash memory of the memory card 40. The encrypted data is serially transmitted between the DSP 30 and the memory card 40. In addition, an external SRAM (Static Random Access Memory) 31 is connected to the DSP 30.

[0026]

A bus interface 32 is connected to the DSP 30. Data is supplied from an external controller (not shown) to the DSP 30 through a bus 33. The external controller controls all operations of the audio system. The external controller supplies data such as a record command or a reproduction command that is generated corresponding to a user's operation through an operation portion to the DSP 30 through the bus interface 32. In addition, the external controller supplies additional information such as image information and character information to the DSP 30 through the bus interface 32. The bus 33 is a bidirectional communication path. Additional information that is read from the memory card 40 is supplied to the external controller through the DSP 30, the bus interface 32, and the bus 33. In reality, the external controller is disposed in for example an amplifying unit of the audio system. In addition, the external controller causes a display portion to display

additional information, the operation state of the recorder, and so forth. The display portion is shared by the audio system. Since data that is exchanged through the bus 33 is not copyright protected data, it is not encrypted.

[0027]

The encrypted audio data that is read from the memory card 40 by the DSP 30 is decrypted by the security IC 20. The audio encoder/decoder IC 10 decodes the encoded data corresponding to the ATRAC3 format. Output data of the audio encoder/decoder 10 is supplied to a D/A converter 18. The D/A converter 18 converts the output data of the audio encoder/decoder 10 into an analog signal. The analog audio signal is supplied to a line output terminal 19.

[0028]

The analog audio signal is supplied to an amplifying unit (not shown) through the line output terminal 19. The analog audio signal is reproduced from a speaker or a head set. The external controller supplies a muting signal to the D/A converter 18. When the muting signal represents a mute-on state, the external controller prohibits the audio signal from being output from the line output terminal 19.

[0029]

Fig. 2 is a block diagram showing the internal structure of the DSP 30. Referring to Fig. 2,

the DSP 30 comprises a core 34, a flash memory 35, an SRAM 36, a bus interface 37, a memory card interface 38, and inter-bus bridges. The DSP 30 has the same function as a microcomputer. The core 34 is equivalent to a CPU. The flash memory 35 stores a program that causes the DSP 30 to perform predetermined processes. The SRAM 36 and the external SRAM 31 are used as a RAM of the recording/reproducing apparatus.

[0030]

The DSP 30 controls a writing process for writing encrypted audio data and additional information to the memory card 40 corresponding to an operation signal such as a record command received through the bus interfaces 32 and 37 and a reading process for reading them therefrom. In other words, the DSP 30 is disposed between the application software side of the audio system that records/reproduces audio data and additional information and the memory card 40. The DSP 30 is operated when the memory card 40 is accessed. In addition, the DSP 30 is operated corresponding to software such as a file system.

[0031]

The DSP 30 manages files stored in the memory card 40 with the FAT system used in conventional personal computers. In addition to the file system, according to the embodiment of the present invention, a management file is used. The management file will be

descried later. The management file is used to manage data files stored in the memory card 40. The management file as the first file management information is used to manage audio data files. On the other hand, the FAT as the second file management information is used to mange all files including audio data files and management files stored in the flash memory of the memory card 40. The management file is stored in the memory card 40. The FAT is written to the flash memory along with the route directory and so forth before the memory card 40 is shipped.

[0032]

According to the embodiment of the present invention, to protect the copyright of data, audio data that has been compressed corresponding to the ATRAC3 format is encrypted. On the other hand, since it is not necessary to protect the copyright of the management file, it is not encrypted. There are two types of memory cards that are an encryption type and a non-encryption type. However, a memory card for use with the recorder/player that records copyright protected data is limited to the encryption type.

[0033]

Fig. 3 is a block diagram showing the internal structure of the memory card 40. The memory card 40 comprises a control block 41 and a flash memory 42 that are structured as a one-chip IC. A

bidirectional serial interface is disposed between the DSP 30 of the recorder/player and the memory card 40. The bidirectional serial interface is composed of ten lines that are a clock line SCK for transmitting a clock signal that is transmitted along with data, a status line SBS for transmitting a signal that represents a status, a data line DIO for transmitting data, an interrupt line INT, two GND lines, two INT lines, and two reserved lines.

[0034]

The clock line SCK is used for transmitting a clock signal in synchronization with data. The status line SBS is used for transmitting a signal that represents the status of the memory card 40. The data line DIO is used for inputting and outputting a command and encrypted audio data. The interrupt line INT is used for transmitting an interrupt signal that causes the memory card 40 to interrupt the DSP 30 of the recorder/player. When the memory card 40 is attached to the recorder/player, the memory card 40 generates the interrupt signal. However, according to the embodiment of the present invention, since the interrupt signal is transmitted through the data line DIO, the interrupt line INT is grounded.

[0035]

A serial/parallel converting, parallel/serial converting, and interface block (S/P, P/S, I/F block)

43 is an interface disposed between the DSP 30 of the recorder/player and the control block 41 of the memory card 40. The S/P, P/S, and IF block 43 converts serial data received from the DSP 30 of the recorder/player into parallel data and supplies the parallel data to the control block 41. In addition, the S/P, P/S, and IF block 43 converts parallel data received from the control block 41 into serial data and supplies the serial data to the DSP 30. When the S/P, P/S, and IF block 43 receives a command and data through the data line DIO, the S/P, P/S, and IF block 43 separates them into these that are normally accessed to the flash memory 42 and those that are encrypted.

[0036]

In the format of which data is transmitted through the data line DIO, after a command is transmitted, data is transmitted. The S/P, P/S, and IF block 43 detects the code of a command and determines whether the command and data are those that are normally accessed or those that are encoded. Corresponding to the determined result, the S/P, P/S, and IF block 43 stores a command that is normally accessed to a command register 44 and stores data that is normally accessed to a page buffer 45 and a write register 46. In association with the write register 46, the memory card 40 has an error correction code encoding circuit 47. The error correction code

encoding circuit 47 generates a redundant code that is an error correction code for data temporarily stored in the page buffer 45.

[0037]

Output data of the command register 44, the page buffer 45, the write register 46, and the error correction code encoding circuit 47 is supplied to a flash memory interface and sequencer (hereinafter, referred to as memory I/F and sequencer) 51. The memory IF and sequencer 51 is an interface disposed between the control block 41 and the flash memory 42 and controls data exchanged therebetween. Data is written to the flash memory 42 through the memory IF and sequencer 51.

[0038]

Contents are written to the flash memory 42 (audio data that has been compressed corresponding to the ATRAC3 format, hereinafter, this audio data is referred to as ATRAC3 data) is encrypted by the security IC 20 of the recorder and the security block 52 of the memory card 40 so as to protect the copyright of the ATRAC3 data. The security block 52 comprises a buffer memory 53, a DES encrypting circuit 54, and a nonvolatile memory 55.

[0039]

The security block 52 of the memory card 40 has a plurality of authentication keys and a unique

storage key for each memory card. The nonvolatile memory 55 stores a key necessary for encrypting data which is unseen from the outer side. For example, a storage key is stored in the nonvolatile memory 55. The security block 52 also has a random number generating circuit. The security block 52 authenticates a specified recorder (it means that using a data format and/or the like are in the same system) and shares a session key therewith. In addition, the security block 52 can be re-encrypted contents with the storage key through the DSE encrypting circuit 54.

[0040]

For example, when the memory card 40 is attached to the recorder, they are mutually authenticated. The security IC 20 of the recorder and the security block 52 of the memory card 40 mutually authenticate. When the recorder has authenticated the attached memory card 40 (a memory card is in the same system) and the memory card 40 has authenticated the recorder (a record is in the same system), they are mutually authenticated. After the mutual authenticating process has been successfully performed, the recorder and the memory card 40 generate respective session keys and share them with each other. Whenever the recorder and the memory card 40 authenticate each other, they generate respective session keys.

[0041]

When contents are written to the memory card 40, the recorder encrypts a contents key with a session key and supplies the encrypted data to the memory card 40. The memory card 40 decrypts the contents key with the session key, re-encrypts the contents key with a storage key, and supplies the contents key to the recorder. The storage key is a unique key for each memory card 40. When the recorder receives the encrypted contents key, the recorder performs a formatting process for the encrypted contents key, and writes the encrypted contents key and the encrypted contents to the memory card 40.

[0042]

Data that is read from the flash memory 42 is supplied to the page buffer 45, the read register 48, and the error correction circuit 49 through the memory IF and the sequencer 51. The error correcting circuit 49 corrects an error of the data stored in the page buffer 45. Output data of the page buffer 45 that has been error-corrected and the output data of the read register 48 are supplied to the S/P, P/S, and IF block 43. The output data of the S/P, P/S, and IF block 43 is supplied to the DSP 30 of the recorder through the above-described serial interface.

[0043]

When data is read from the memory card 40,

the contents key encrypted with the storage key and the contents encrypted with the block key are read from the flash memory 42. The security block 52 decrypts the contents key with the storage key. The security block 52 re-encrypts the decrypted content key with the session key and transmits the re-encrypted contents key to the recorder. The recorder decrypts the contents key with the received session key and generates a block key with the decrypted contents key. The recorder successively decrypts the encrypted ATRAC3 data.

[0044]

A config. ROM 50 is a configuration ROM that stores partition information, various types of attribute information, and so forth of the memory card 40. The memory card 40 also has an erase protection switch 60. When the switch 60 is in the erase protection position, even if a command that causes the memory card 40 to erase data stored in the flash memory 42 is supplied from the recorder side to the memory card 40, the memory card 40 is prohibited from erasing the data stored in the flash memory 42. An OSC cont. 61 is an oscillator that generates a clock signal that is the reference of the timing of the process of the memory card 40.

[0045]

Fig. 4 is a schematic diagram showing the hierarchy of the processes of the file system of the

computer system that uses a memory card as a storage medium. On the hierarchy, the top hierarchical level is an application process layer. The application process layer is followed by a file management process layer, a logical address management layer, a physical address management layer, and a flash memory access layer. In the hierarchical structure, the file management process layer is the FAT file system. Physical addresses are assigned to individual blocks of the flash memory. The relation between the blocks of the flash memory and the physical addresses thereof does not vary. Logical addresses are addresses that are logically handled on the file management process layer.

[0046]

Fig. 5 is a schematic diagram showing the physical structure of data handled in the flash memory 42 of the memory card 40. In the memory 42, a data unit (referred to as segment) is divided into a predetermined number of blocks (fixed length). One block is divided into a predetermined number of pages (fixed length). In the flash memory, data is erased as each block at a time. Data is written to the flash memory 42 or read therefrom as a page at a time. The size of each block is the same. Likewise, the size of each page is the same. One block is composed of page 0 to page m. For example, one block has a storage

capacity of for example 8 KB (kilobytes) or 16 KB. One page has a storage capacity of 512 B (bytes). When one block has a storage capacity of 8 KB, the total storage capacity of the flash memory 42 is 4 MB (512 blocks) or 8 MB (1024 blocks). When one block has a storage capacity of 16 KB, the total storage capacity of the flash memory 42 is 16 MB (1024 blocks), 32 MB (2048 blocks), or 64 MB (4096 blocks).

[0047]

One page is composed of a data portion of 512 bytes and a redundant portion of 16 bytes. The first three bytes of the redundant portion is an overwrite portion that is rewritten whenever data is updated. The first three bytes successively contain a block status area, a page status area, and an update status area. The remaining 13 bytes of the redundant portion are fixed data that depends on the contents of the data portion. The 13 bytes contain a management flag area (1 byte), a logical address area (2 bytes), a format reserve area (5 bytes), a dispersion information ECC area (2 bytes), and a data ECC area (3 bytes). The dispersion information ECC area contains redundant data for an error correction process against the management flag area, the logical address area, and the format reserve area. The data ECC area contains redundant data for an error correction process against 512-byte data.

[0048]

The management flag area contains a system flag (1: user block, 0: boot block), a conversion table flag (1: invalid, 0: table block), a copy prohibition flag (1: OK, 0: NG), and an access permission flag (1: free, 0: read protect).

[0049]

The first two blocks - blocks 0 and 1 are boot blocks. The block 1 is a backup of the block 0. The boot blocks are top blocks that are valid in the memory card. When the memory card is attached to the recorder, the boot blocks are accessed at first. The remaining blocks are user blocks. Page 0 of the boot block contains a header area, a system entry area, and a boot and attribute information area. Page 1 of the boot block contains a prohibited block data area. Page 2 of the boot block contains a CIS (Card Information Structure)/IDI (identify Drive Information) area.

[0050]

The header area of the boot block contains a boot block ID and the number of effective entries. The system entries are the start position of prohibited block data, the data size thereof, the data type thereof, the data start position of the CIS/IDI area, the data size thereof, and the data type thereof. The boot and attribute information contains the memory card type (read only type, rewritable type, or hybrid type),

the block size, the number of blocks, the number of total blocks, the security/non-security type, the card fabrication data (date of fabrication), and so forth.

[0051]

Since the flash memory has a restriction for the number of rewrite times due to the deterioration of the insulation film, it is necessary to prevent the same storage area (block) from being concentratedly accessed. Thus, when data at a particular logical address stored at a particular physical address is rewritten, updated data of a particular block is written to a non-used block rather than the original block. Thus, after data is updated, the relation between the logical address and the physical address changes. Consequently, the same block is prevented from being concentratedly accessed by executing this process (this process is referred to as swap process). Thus, the service life of the flash memory can be prolonged.

[0052]

The logical address associates with data written to the block. Even if the block of the original data is different from the block of updated data, the address on the FAT does not change. Thus, the same data can be properly accessed. However, since the swap process is performed, a conversion table that correlates logical addresses and physical addresses is

required (this table is referred to as logical-physical address conversion table). With reference to the logical-physical address conversion table, a physical address corresponding to a logical address designated on the FAT is obtained. Thus, a block designated with a physical address can be accessed.

[0053]

The DSP 30 stores the logical-physical address conversion table in the SRAM. When the storage capacity of the RAM is small, the logical-physical address conversion table can be stored to the flash memory. The logical-physical address conversion table correlates logical addresses (2 bytes) sorted in the ascending order with physical addresses (2 bytes). Since the maximum storage capacity of the flash memory is 128 MB (8192 blocks), 8192 addresses can be assigned with two bytes. The logical-physical address conversion table is managed for each segment. Thus, the size of the logical-physical address conversion table is proportional to the storage capacity of the flash memory. When the storage capacity of the flash memory is 8 MB (two segments), two pages are used as the logical-physical address conversion table for each of the segments. When the conversion table is stored in the flash memory, a predetermined one bit of the management flag area in the redundant portion in each page represents whether or not the current block is a

block containing the logical-physical address conversion table.

[0054]

The above-described memory card can be used with the FAT file system of a personal computer system as with the disc shaped record medium. The flash memory has an IPL area, a FAT area, and a route directory area (not shown in Fig. 5). The IPL area contains the address of a program to be initially loaded to the memory of the recorder. In addition, the IPL area contains various types of memory information. The FAT area contains information with respect to blocks (clusters). The FAT has defined unused blocks, next block number, defective blocks, and last block number. The route directory area contains directory entries that are a file attribute, an update date [day, month, year], file size, and so forth.

[0055]

According to the embodiment of the present invention, in addition to the file management system defined in the format of the memory card 40, the management file is used for managing tracks and parts of music files. The management file is recorded to a user block of the flash memory 42 of the memory card 40. Thus, as will be described later, even if the FAT of the memory card 40 is destroyed, a file can be recovered.

[0056]

The management file is generated by the DSP 30. When the power of the recorder is turned on, the DSP 30 determines whether or not the memory card 40 has been attached to the recorder. When the memory card has been attached, the DSP 30 authenticates the memory card 40. When the DSP 30 has successfully authenticated the memory card 40, the DSP 30 reads the boot block of the flash memory 42. Thus, the DSP 30 reads the physical-logical address conversion table and stores the read data to the SRAM. The FAT and the route directory have been written to the flash memory of the memory card 40 before the memory card 40 is shipped. When data is recorded to the memory card 40, the management file is generated.

[0057]

In other words, a record command issued by the remote controller of the user or the like is supplied to the DSP 30 from the external controller through the bus and the bus interface 32. The encoder/decoder IC 10 compresses the received audio data and supplies the resultant ATRAC3 data to the security IC 20. The security IC 20 encrypts the ATRAC3 data. The encrypted ATRAC3 data is recorded to the flash memory 42 of the memory card 40. Thereafter, the FAT and the management file are updated. Whenever a file is updated (in reality, whenever the recording

process of audio data is completed), the FAT and the management file stored in the SRAMs 31 and 36 are rewritten. When the memory card 40 is detached or the power of the recorder is turned off, the FAT and the management file that are finally supplied from the SRAMs 31 and 36 are recorded to the flash memory 42. Alternatively, whenever the recording process of audio data is completed, the FAT and the management file written in the flash memory 42 may be rewritten. When audio data is edited, the contents of the management file are updated.

[0058]

In the data structure according to the embodiment, additional information is contained in the management file. The additional information is updated and recorded to the flash memory 42. In another data structure of the management file, an additional information management file is generated besides the track management file. The additional information is supplied from the external controller to the DSP 30 through the bus and the bus interface 32. The additional information is recorded to the flash memory 42 of the memory card 40. Since the additional information is not supplied to the security IC 20, it is not encrypted. When the memory card 40 is detached from the recorder or the power thereof is turned off, the additional information is written from the SRAM of

the DSP 30 to the flash memory 42.

[0059]

Fig. 6 is a schematic diagram showing the file structure of the memory card 40. As the file structure, there are a still picture directory, a moving picture directory, a voice directory, a control directory, and a music (HIFI) directory. According to the embodiment, music programs are recorded and reproduced. Next, the music directory will be described. The music directory has two types of files. The first type is a reproduction management file BLIST.MSF (hereinafter, referred to as PBLIST). The other type is an ATRAC3 data file A3Dnnnn.MSA that stores encrypted music data. The music directory can stores up to 400 ATRAC3 data files (namely, 400 music programs). ATRAC3 data files are registered to the reproduction management file and generated by the recorder.

[0060]

Fig. 7 is a schematic diagram showing the structure of the reproduction management file. Fig. 8 is a schematic diagram showing the file structure of one ATRAC3 data file. The reproduction management file is a fixed-length file of 16 KB. An ATRAC3 data file is composed of an attribute header and an encrypted music data area for each music program. The attribute data has a fixed length of 16 KB. The structure of the

attribute header is similar to that of the reproduction management file.

[0061]

The reproduction management file shown in Fig. 7 is composed of a header, a memory card name NM-1S (for one byte code), a memory card name NM2-S (for two byte code), a program reproduction sequence table TRKTBL, and memory card additional information INF-S. The attribute header at the beginning of the data file is composed of a header, a program name NM1 (for one byte code), a program name NM2 (for two byte code), track information TRKINF (such as track key information), part information PRTINF, and track additional information INF. The header contains information of the number of total parts, the attribute of the name, the size of the additional information, and so forth.

[0062]

The attribute data is followed by ATRAC3 music data. The music data is block-segmented every 16 KB. Each block starts with a header. The header contains an initial value for decrypting encrypted data. Only music data of an ATRAC3 data file is encrypted. Thus, other data such as the reproduction management file, the header, and so forth are not encrypted.

[0063]

Next, with reference to Figs. 9, the relation between music programs and ATRAC3 data files will be described. One track is equivalent to one music program. In addition, one music program is composed of one ATRAC3 data (see Fig. 8). The ATRAC3 data file is audio data that has been compressed corresponding to the ATRAC3 format. The ATRAC3 data file is recorded as a cluster at a time to the memory card 40. One cluster has a capacity of 16 KB. A plurality of files are not contained in one cluster. The minimum data erase unit of the flash memory 42 is one block. In the case of the memory card 40 for music data, a block is a synonym of a cluster. In addition, one cluster is equivalent to one sector.

[0064]

One music program is basically composed of one part. However, when a music program is edited, one music program may be composed of a plurality of parts. A part is a unit of data that is successively recorded. Normally, one track is composed of one part. The connection of parts of a music program is managed with part information PRTINF in the attribute header of each music program. In other words, the part size is represented with part size PRTSIZE (4 bytes) of the part information PRTINF. The first two bytes of the part size PRTSIZE represents the number of total clusters of the current part. The next two bytes

represent the positions of the start sound unit (SU) and the end sound unit (SU) of the beginning and last clusters, respectively. Hereinafter, a sound unit is abbreviated as SU. With such a part notation, when music data is edited, the movement of the music data can be suppressed. When music data is edited for each block, although the movement thereof can be suppressed, the edit unit of a block is much larger than the edit unit of a SU.

[0065]

SU is the minimum unit of a part. In addition, SU is the minimum data unit in the case that audio data is compressed corresponding to the ATRAC3 format. 1 SU is audio data of which data of 1024 samples at 44.1 kHz (1024 x 16 bits x 2 channels) is compressed to data that is around 10 times smaller than that of original data. The duration of 1 SU is around 23 msec. Normally, one part is composed of several thousand SU. When one cluster is composed of 42 SU, one cluster allows a sound of one second to be generated. The number of parts composing one track depends on the size of the additional information. Since the number of parts is obtained by subtracting the header, the program name, the additional data, and so forth from one block, when there is no additional information, the maximum number of parts (645 parts) can be used.

[0066]

Fig. 9 is a schematic diagram showing the file structure in the case that two music programs of a CD or the like are successively recorded. The first program (file 1) is composed of for example five clusters. Since one cluster cannot contain two files of the first program and the second program, the file 2 starts from the beginning of the next cluster. Thus, the end of the part 1 corresponding to the file 1 is in the middle of one cluster and the remaining area of the cluster contains no data. Likewise, the second music program (file 2) is composed of one part. In the case of the file 1, the part size is 5. The first cluster starts at 0-th SU. The last cluster ends at 4-th SU.

[0067]

There are four types of edit processes that are a divide process, a combine process, an erase process, and a move process. The divide process is performed to divide one track into two portions. When the divide process is performed, the number of total tracks increases by one. In the divide process, one file is divided into two files on the file system. Thus, in this case, the reproduction management file and the FAT are updated. The combine process is performed to combine two tracks into one track. When the combine process is performed, the number of total tracks decreases by one. In the combine process, two

files are combined into one file on the file system. Thus, when the combine process is performed, the reproduction management file and the FAT are updated. The erase process is performed to erase a track. The track numbers after the track that has been erased decrease one by one. The move process is performed to change the track sequence. Thus, when the erase process or the move process is performed, the reproduction management file and the FAT are updated.

[0068]

Fig. 9 is a schematic diagram showing the combined result of two programs (file 1 and file 2) shown in Fig. 10. As a result of the combine process, the combined file is composed of two parts. Fig. 11 is a schematic diagram showing the divided result of which one program (file 1) is divided in the middle of the cluster 2. By the divide process, the file 1 is composed of clusters 0, 1, and the beginning portion of cluster 2. The file 2 is composed of the end portion of cluster 2 and clusters 3 and 4.

[0069]

As described above, according to the embodiment of the present invention, since the part notation is defined, as the combined result (see Fig. 10), the start position of the part 1, the end position of the part 1, and the end portion of the part 2 can be defined with SU. Thus, to pack the space due to the

combined result, it is not necessary to move the music data of the part 2. In addition, as the divided result (see Fig. 10C), it is not necessary to move data and pack the space at the beginning of the file 2.

[0070]

Fig. 12 is a schematic diagram showing the detailed data structure of the reproduction management file PBLIST. Figs. 13A and 13B show a header portion and the remaining portion of the reproduction management file PBLIST. The size of the reproduction management file is one cluster (one block = 16 KB). The size of the header shown in Fig. 13A is 32 bytes. The rest of the reproduction management file PBLIST shown in Fig. 13B contains a name NM1-S area (256 bytes) (for the memory card), a name NM2-S area (512 bytes), a contents key area, a MAC area, an S-YMDhms area, a reproduction sequence management table TRKTBL area (800 bytes), a memory card additional information INF-S area (14720 bytes), and a header information redundant area. The start positions of these areas are defined in the reproduction management file.

[0071]

The first 32 bytes of (0x0000) to (0x0010) shown in Fig. 13A are used for the header. In the file, 16-byte areas are referred to as slots. Referring to Fig. 13A, the header are placed in the first and second slots. In order, the data having the

following meanings, function, values are supplied to those headers from the first header. The header contains the following areas. An area denoted by "Reserved" is an undefined area. Normally, in a reserved area, a null (0x00) is written. However, even if any data is written to a reserved area, the data written in the reserved is ignored. In a future version, some reserved areas may be used. In addition, data is prohibited from being written to a reserved area. When an option area is not used, it is treated as a reserved area.

[0072]

= BLKID-TL0 (4 bytes)

Meaning: BLOCKID FILE ID

Function: Identifies the top of the reproduction management file.

Value: Fixed value = "TL = 0" (for example, 0x544C2D30)

= MCode (2 bytes)

Meaning: MAKER CODE

Function: Identifies the maker and model of the recorder

Value: High-order 10 bits (Maker code); low-order 6 bits (model code).

= REVISION (4 bytes)

Meaning: Number of rewrite times of PBLIST

Function: Increments whenever the

reproduction management file is rewritten.

Value: Starts at 0 and increments by 1.

= S-YMDhms (4 bytes) (Option)

Meaning: Year, month, day, hour, minute, and second recorded by the recorder with a reliable clock.

Function: Identifies the last recorded date and time.

Value: bits 25 to 31: Year 0 to 99 (1980 to 2079)

bits 21 to 24: Month 0 to 12

bits 16 to 20: Day 0 to 31

bits 11 to 15: Hour 0 to 23

bits 05 to 10: Minute 0 to 59

bits 00 to 04: Second 0 to 29 (two bit interval)

[0073]

= SY1C+L (2 bytes)

Meaning: Attribute of name (one byte code) of memory card written in NM1-S area.

Function: Represents the character code and the language code as one byte code.

Value: Character code (C): High-order one byte

00: Non-character code, binary number

01: ASCII (American Standard Code for Information Interchange)

02: ASCII+KANA

03: Modified 8859-1

81: MS-JIS

82: KS C 5601-1989

83: GB (Great Britain) 2312-80

90: S-JIS (Japanese Industrial Standards) (for Voice)

[0074]

Language code (L): Low-order one byte

Identifies the language based on EBU Tech 3258 standard.

00: Not set

08: German

09: English

0A: Spanish

0F: French

15: Italian

1D: Dutch

65: Korean

69: Japanese

75: Chinese

When data is not recorded, this area is all 0.

[0075]

= SN2C+L (2 bytes)

Meaning: Attribute of name of memory card in NM2-S area.

Function: Represents the character code and

the language coded as one byte code.

Value: Same as SN1C+L

= SINFSIZE (2 bytes)

Meaning: Total size of additional information of memory card in INF-S area.

Function: Represents the data size as an increment of 16 bytes. When data is not recorded, this area is all 0.

Value: Size: 0x0001 to 0x39C (924)

= T-TRK (2 bytes)

Meaning: TOTAL TRACK NUMBER

Function: Represents the number of total tracks.

Value: 1 to 0x0190 (Max. 400 tracks)

When data is recorded, this area is all 0.

= VerNo (2 bytes)

Meaning: Format version number

Function: Represents the major version number (high order one byte) and the minor version number (low order one byte).

Value: 0x0100 (Ver 1.0)

0x0203 (Ver 2.3)

[0076]

Next, areas (see Fig. 13B) that preceded by the header will be described.

[0077]

= NM1-S

Meaning: Name of memory card (as one byte code)

Function: Represents the name of the memory card as one byte code (max. 256). At the end of this area, an end code (0x00) is written. The size is calculated from the end code. When data is not recorded, null (0x00) is recorded from the beginning (0x0020) of this area for at least one byte.

Value: Various character code

= NM2-S

Meaning: Name of memory card (as two byte code)

Function: Represents the name of the memory card as two byte code (max. 512). At the end of this area, an end code (0x00) is written. The size is calculated from the end code. When data is not recorded, null (0x00) is recorded from the beginning (0x0120) of this area for at least two bytes.

Value: Various character code

[0078]

= CONTENTS KEY

Meaning: Value for music program. Protected with MG(M) and stored. Same as CONTENTS KEY.

Function: Used as a key necessary for calculating MAC of S-YMDhms.

Value: 0 to 0xFFFFFFFFFFFFFFFF

= MAC

Meaning: Forged copyright information check value

Function: Represents the value generated with S-YMDhms and CONTENTS KEY.

Value: 0 to 0xFFFFFFFFFFFFFFFF

[0079]

= TRK-nnn

Meaning: SQN (sequence) number of ATRAC3 data file reproduced.

Function: Represents FNo of TRKINF.

Value: 1 to 400 (0x190)

When there is no track, this area is all 0.

= INF-S

Meaning: Additional information of memory card (for example, information with respect to photos, songs, guides, etc.)

Function: Represents variable length additional information with a header. A plurality of types of additional information may be used. Each of the types of additional information has an ID and a data size. Each additional information area including a header is composed of at least 16 bytes and a multiple of 4 bytes. For details, see the following section.

Value: Refer to the section of "Data Structure of Additional Information".

= S-YMDhms (4 bytes) (Option)

Meaning: Year, month, day, hour, minute, and second recorded by the recorder with a reliable clock.

Function: Identifies the last recorded date and time. In this case of EMD, this area is mandatory.

Value: bits 25 to 31: Year 0 to 99 (1980 to 2079)

bits 21 to 24: Month 0 to 12

bits 16 to 24: Day 0 to 31

bits 11 to 15: Hour 0 to 23

bits 05 to 10: Minute 0 to 59

bits 00 to 04: Second 0 to 29 (two second interval)

[0080]

As the last slot of the reproduction management file, the same BLKID-TL0, MCode, and REVISION as those in the header are written.

[0081]

While data is being recorded to a memory card, it may be mistakenly or accidentally detached or the power of the recorder may be turned off. When such an improper operation is performed, a defect should be detected. As described above, the REVISION area is placed at the beginning and end of each block. Whenever data is rewritten, the value of the REVISION area is incremented. If a defect termination takes place in the middle of a block, the value of the REVISION area at the beginning of the block does not

match the value of the REVISION area at the end of the block. Thus, such a defect termination can be detected. Since there are two REVISION areas, the abnormal termination can be detected with a high probability. When an abnormal termination is detected, an alarm such as an error message is generated.

[0082]

In addition, since the fixed value BLKID-TL0 is written at the beginning of one block (16 KB), when the FAT is destroyed, the fixed value is used as a reference for recovering data. In other words, with reference to the fixed value, the type of the file can be determined. Since the fixed value BLKID-TL0 is redundantly written at the header and the end portion of each block, the reliability can be secured. Alternatively, the same reproduction management file can be redundantly recorded.

[0083]

The data amount of an ATRAC3 data file is much larger than that of the track information management file. In addition, as will be described later, a block number BLOCK SERIAL is added to ATRAC3 data file. However, since a plurality of ATRAC3 files are recorded to the memory card, to prevent them from become redundant, both CONNUM0 and BLOCK SERIAL are used. Otherwise, when the FAT is destroyed, it will be difficult to recover the file. In other words, one

ATRAC3 data file may be composed of a plurality of blocks that are dispersed. To identify blocks of the same file, CONNUM0 is used. In addition, to identify the order of blocks in the ATRAC3 data file, BLOCK SERIAL is used.

[0084]

Likewise, the maker code (Mcode) is redundantly recorded at the beginning and the end of each block so as to identify the maker and the model in such a case that a file has been improperly recorded in the state that the FAT has not been destroyed.

[0085]

Fig. 13C is a schematic diagram showing the structure of the additional information data. The additional information is composed of the following header and variable length data. The header has the following areas.

[0086]

= INF

Meaning: FIELD ID

Function: Represents the beginning of the additional information (fixed value).

Value: 0x69

= ID

Meaning: Additional information key code

Function: Represents the category of the additional information.

Value: 0 to 0xFF

= SIZE

Meaning: Size of individual additional information

Function: Represents the size of each type of additional information. Although the data size is not limited, it should be at least 16 bytes and a multiple of 4 bytes. The rest of the data should be filled with null (0x00).

Value: 16 to 14784 (0x39C0)

= MCode

Meaning: MAKER CODE

Function: Identifies the maker and model of the recorder.

Value: High-order 10 bits (maker code), low-order 10 bits (machine code).

= C+L

Meaning: Attribute of characters in data area starting from byte 12.

Function: Represents the character code and the language code as one byte code.

Value: Same as SNC+L

= DATA

Meaning: Individual additional information

Function: Represents each type of additional information with variable length data. Real data always starts from byte 12. The length (size) of the

real data should be at least 4 bytes and a multiple of 4 bytes. The rest of the data area should be filled with null (0x00).

Value: Individually defined corresponding to the contents of each type of additional information.

[0087]

Fig. 14 is a table that correlates key code values (0 to 63 of additional information and types thereof. Key code values (0 to 31) are assigned to music character information. Key code values (32 to 63) are assigned to URLs (Uniform Resource Locator) (web information). The music character information and URL information contain character information of the album title, the artist name, the CM, and so forth as additional information.

[0088]

Fig. 15 is a table that correlates key code values (64 to 127) of additional information and types thereof. Key code values (64 to 95) are assigned to paths/others. Key code values (96 to 127) are assigned to control/numeric data. For example, ID = 98 represents TOC-ID as additional information. TOC-ID represents the first music program number, the last music program number, the current program number, the total performance duration, and the current music program duration corresponding to the TOC information of a CD (Compact Disc).

[0089]

Fig. 16 is a table that correlates key code values (128 to 159) of additional information and types thereof. Key code values (128 to 159) are assigned to synchronous reproduction information. In Fig. 15, EMD stands for electronic music distribution.

[0090]

Next, with reference to Fig. 17, real examples of additional information will be described. As with Fig. 13C, Fig. 17A shows the data structure of the additional information. In Fig. 17B, key code ID = 3 (artist name as additional information). SIZE = 0x1C (28 bytes) representing that the data length of additional information including the header is 28 bytes; C+L representing that character code C = 0x01 (ASCII) and language code L = 0x09 (English). Variable length data after byte 12 represents one byte data "SIMON & GRAFUNKEL" as artist name. Since the data length of the additional information should be a multiple of 4 bytes, the rest is filled with (0x00).

[0091]

In Fig. 17C, key code ID = 97 representing that ISRC (International Standard Recording Code: Copyright code) as additional information. SIZE = 0x14 (20 bytes) representing that the data length of the additional information is 20 bytes. C = 0x00 and L = 0x00 representing that characters and language have not

been set. Thus, the data is binary code. The variable length data is eight-byte ISRC code representing copyright information (nation, copyright owner, recorded year, and serial number).

[0092]

In Fig. 17D, key code ID =is 97 representing recorded date and time as additional information. SIZE = 0 x 10 (16 bytes) representing that the data length of the additional information is 16 bytes. C = 0x00 and L = representing that characters and language have not been set. The variable length data is four-byte code (32 bit) representing the recorded date and time (year, month, day, hour, minute, second).

[0093]

In Fig. 17E, key code ID = 107 representing a reproduction log as additional information. SIZE = 0x10 (16 bytes) representing that the data length of the additional information is 16 bytes. C = 0x00 and L = 0x00 representing that characters and language have not been set. The variable length data is a four-byte code representing a reproduction log (year, month, day, hour, minute, second). When the recorder has a reproduction log function, it records data of 16 bytes whenever it reproduces music data.

[0094]

Fig. 18 is a schematic diagram showing a data arrangement of ATRAC3 data file A3Dnnnn in the case

that 1 SU is N bytes (for example, N = 384 bytes). Fig. 18 shows an attribute header (1 block) of a data file and a music data file (1 block). Fig. 18 shows the first byte (0x0000 to 0x7FF0) of each slot of the two blocks (16 x 2 = 32 kbytes). As shown in Fig. 18, the first 32 bytes of the attribute header are used as a header; 256 bytes are used as a music program area NM1 (256 bytes); and 512 bytes are used as a music program title area NM2 (512 bytes). The header of the attribute header contains the following areas.

[0095]

= BLKID-HD0 (4 bytes)

Meaning: BLOCKID FIELD ID

Function: Identifies the top of an ATRA3 data file.

Value: Fixed value = "HD = 0" (For example, 0x48442D30)

= MCode (2 bytes)

Meaning: MAKER CODE

Function: Identifies the maker and model of the recorder

Value: High-order 10 bits (maker code); low-order 6 bits (machine code)

= BLOCK SERIAL (4 bytes)

Meaning: Track serial number

Function: Starets from 0 and increments by 1. Even if a music program is edited, this value does

not vary.

Value: 0 to 0xFFFFFFFF.

[0096]

= N1C+L (2 bytes)

Meaning: Represents the attribute of data (NM1) of a track (music program title).

Function: Represent the character code and language code of NM1 as one byte code.

Value: Same as SN1C+L

= N2C+L (2 bytes)

Meaning: Represents the attribute of data (NM2) of a track (music program title).

Function: Represent the character code and language code of NM1 as one byte code.

Value: Same as SN1C+L

= INFSIZE (2 bytes)

Meaning: Total size of additional information of current track.

Function: Represents the data size as a multiple of 16 bytes. When data is not recorded, this area should be all 0.

Value: 0x0000 to 0x3C6 (966)

= T-PRT (2 bytes)

Meaning: Number of total bytes

Function: Represents the number of parts that composes the current track. Normally, the value of T-PRT is 1.

Value: 1 to 285 (645 dec).

= T-SU (4 bytes)

Meaning: Number of total SU.

Function: Represents the total number of SU in one track that is equivalent to the program performance duration.

Value: 0x01 to 0x001FFFFF

= INX (2 bytes) (Option)

Meaning: Relative position of INDEX

Function: Used as a pointer that represents the top of a representative portion of a music program. The value of INX is designated with a value of which the number of SU is divided by 4 as the current position of the program. This value of INX is equivalent to 4 times larger than the number of SU (around 93 msec).

Value: 0 to 0xFFFF (max, around 6084 sec)

= XT (2 bytes) (Option)

Meaning: Reproduction duration of INDEX

Function: Designates the reproduction duration designated by INX-nnn with a value of which the number of SU is divided by 4. The value of INDEX is equivalent to four times larger than the normal SU (around 93 msec).

Value: 0x0000 (no setting); 0x01 to 0xFFFE (up to 6084 sec); 0xFFFF (up to end of music program).

[0097]

Next, the music program title areas NM1 and NM2 will be described.

[0098]

= NM1

Means: Character string of music program title

Function: Represents a music program title as one byte code (up to 256 characters) (variable length). The title area should be completed with an end code (0x00). The size should be calculated from the end code. When data is not recorded, null (0x00) should be recorded from the beginning (0x0020) of the area for at least one byte.

Value: Various character codes

= NM2

Means: Character string of music program title

Function: Represents a music program title as two byte code (up to 512 characters) (variable length). The title area should be completed with an end code (0x00). The size should be calculated from the end code. When data is not recorded, null (0x100) should be recorded from the beginning (0x0120) of the area for at least two bytes.

[0099]

Value: Various character codes

Data of 80 bytes starting from the fixed

position (0x320) of the attribute header is referred to as track information area TRKINF. This area is mainly used to totally manage the security information and copy control information. Fig. 19 shows a part of TRKINF. The area TRKINF contains the following areas.

[0100]

= CONTENTS KEY (8 bytes)

Meaning: Value for each music program. The value of CONTENTS KEY is protected in the security block of the memory card and then stored.

Function: Used as a key for reproducing a music program. It is used to calculate the value of MAC.

Value: 0 to 0xFFFFFFFFFFFFFFFF

= MAC (8 bytes)

Meaning: Forged copyright information check value Function: Represents the value generated with a plurality of values of TRKINF including contents cumulation numbers and a secret sequence number.

The secret sequence number is a sequence number recorded in the secret area of the memory card. A non-copyright protection type recorder cannot read data from the secret area of the memory card. On the other hand, a copyright protection type recorder and a computer that operates with a program that can read data from a memory card can access the secret area.

[0101]

= A (1 byte)

Meaning: Attribute of part.

Function: Represents the information of such as compression mode of a part.

Value: The details will be described in the following (see Fig. 21).

Next, the value of the area A will be described. In the following description, monaural mode (N = 0 or 1) is defined as a special joint mode of which bit 7 = 1, sub signal = 0, main signal = (L+R). A non-copyright protection type player may ignore information of bits 2 and 1.

[0102]

Bit 0 of the area A represents information of emphasis on/off state. Bit 1 of the area A represents information of reproduction skip or normal reproduction. Bit 2 of the area A represents information of data type such as audio data, FAX data, or the like. Bit 3 of the area A is undefined. By a combination of bits 4, 5, and 6, mode information of ATRAC3 is defined as shown in Fig. 20. In other words, N is a mode value of 3 bits. For five types of modes that are monaural (N = 0 or 1), LP (N = 2), SP (N = 4), EX (N = 5), and HQ (N = 7), record duration (64 MB memory card only), data transmission rate, and the number of SU per block are listed. The number of bytes of 1 SU depends on each mode. The number of bytes of 1

SU in the monaural mode is 136 bytes. The number of bytes of 1 SU in the LP mode is 192 bytes. The number of bytes of 1 SU in the SP mode is 304 bytes. The number of bytes of 1 SU in the EX mode is 384 bytes. The number of bytes of 1 SU in the HQ mode is 512 bytes. Bit 7 of the area A represents ATRAC3 modes (0: Dual, 1: Joint).

[0103]

For example, an example of which a 64 MB memory card is used in the SP mode will be described. A 64-MB memory card has 3968 blocks. In the SP mode, since 1 SU is 304 bytes, one block has 53 SU. 1 SU is equivalent to (1024/44100) seconds. Thus, one block is (1024/44100) x 53 x (3968 - 10) = 4863 seconds = 81 minutes. The transmission rate is (44100/1024) x 304 x 8 = 104737 bps.

[0104]

= LT (one byte)

Meaning: Reproduction restriction flag (bits 7 and 6) and security partition (bits 5 to 0).

Function: Represents a restriction of the current track.

Value: bit 7: 0 = no restriction, 1 = restriction

bit 6: 0 = not expired, 1 = expired

bits 5 to 0: security partition (reproduction

prohibited other than 0)

= FNo (2 bytes)

Meaning: File number.

Function: Represents the initially recorded track number that designates the position of the MAC calculation value recorded in the secret area of the memory card.

Value: 1 to 0x190 (400)

= MG(D) SERIAL-nnn (16 bytes)

Meaning: Represents the serial number of the security block (security IC 20) of the recorder.

Function: Unique value for each recorder

Value: 0 to 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

= CONNUM (4 bytes)

Meaning: Contents cumulation number

Function: Represents a unique value cumulated for each music program. The value is managed by the security block of the recorder. The upper limit of the value is 2^{32} that is 4,200,000,000. Used to identify a recorded program.

[0105]

Value: 0 to 0xFFFFFFFF

[0106]

YMDhms-S (4 bytes) (Option)

Meaning: Reproduction start date and time of track with reproduction restriction

Function: Represents the date and time at which data reproduction is permitted with EMD.

Value: Same as the notation of date and time of other areas

= YMDhms-E (4 bytes) (Option)

Meaning: Reproduction end date and time of track with reproduction restriction

Function: Represents the date and time at which data reproduction is expired with EMD.

Value: Same as the notation of date and time of other areas

= MT (1 byte) (Option)

Meaning: Maximum value of number of permitted reproduction times

Function: Represents the maximum number of reproduction times designated by EMD.

Value: 1 to 0xFF. When not used, the value of the area MT is 00.

= CT (1 byte) (Option)

Meaning: Number of reproduction times

Function: Represents the number of reproduction times in the number of permitted reproduction times. Whenever data is reproduced, the value of the area CT is decremented.

Value: 0x00 to 0xFF. When not used, the value of the area CT is 0x00. When bit 7 of the area LT is 1 and the value of the area CT is 00, data is

prohibited from being reproduced.

[0107]

= CC (1 byte)

Meaning: COPY CONTROL

Function: Controls the copy operation.

Value: As shown Fig. 22, bits 6 and 7 represent copy control information. bits 4 and 5 represent copy control information of a high speed digital copy operation. bits 2 and 3 represent a security block authentication level. bits 0 and 1 are undefined.

Example of CC:

(bits 7 and 6)

11: Unlimited copy operation permitted

01: copy prohibited

00: one time copy operation permitted

(bits 3 and 2)

00: analog/digital input recording

MG authentication level is 0.

When digital record operation using data from a CD is performed, (bits 7 and 6): 00 and (bits 3 and 2): 00.

= CN (1 byte) (Option)

Meaning: Number of permitted copy times in high speed serial copy management system

Function: Extends the copy permission with the number of copy times, not limited to one time copy

permission and copy free permission. Valid only in first copy generation. The value of the area CN is decremented whenever the copy operation is performed.

Value"

00: Copy prohibited

01 to 0xFE: Number of times

0xFF: Unlimited copy times

[0108]

The track information area TRKINF is followed by a 24-byte part management information area (PRTINF) starting from 0x0370. When one track is composed of a plurality of parts, the values of areas PRTINF of the individual parts are successively arranged on the time axis. Fig. 22 shows a part of the area PRTINF. Next, areas in the area PRTINF will be described in the order of the arrangement.

[0109]

= PRTSIZE (4 bytes)

Meaning: Part size

Function: Represents the size of a part. Cluster: 2 bytes (highest position), start SU: 1 byte (upper), end SU: 1 byte (lowest position).

Value: cluster: 1 to 0x1F40 (8000)

start SU: 0 to 0xA0 (160)

end SU: 0 to 0xA0 (16) (Note that SU starts from 0.)

= PRTKEY (8 bytes)

Meaning: Part encrypting value

Function: Encrypts a part. Initial value = 0. Note that edit rules should be applied.

Value: 0 to 0xFFFFFFFFFFFFFFFF

= CONNUM0 (4 bytes)

Meaning: Initially generated contents cumulation number key

Function: Uniquely designates an ID of contents.

Value: Same value as the value of the contents cumulation number initial value key

[0110]

As shown in Fig. 18, the attribute header of an ATRAC3 data file contains additional information INF. The additional information is the same as the additional information INF-S (see Figs. 12 and 13B) of the reproduction management file except that the start position is not fixed. The last byte position (a multiple of four bytes) at the end of one or a plurality of parts is followed by data of the additional information INF.

[0111]

= INF

Meaning: Additional information with respect to track

Function: Represents variable length additional information with a header. A plurality of

different types of additional information may be arranged. Each of additional information areas has an ID and a data size. Each additional information area is composed of at least 16 bytes and a multiple of 4 bytes.

Value: Same as additional information INF-S of reproduction management file

[0112]

The above-described attribute header is followed by data of each block of an ATRAC3 data file. As shown in Fig. 24, a header is added for each block. Next, data of each block will be described.

[0113]

= BLKID-A3D (4 bytes)

Meaning: BLOCKID FILE ID

Function: Identifies the top of ATRAC3 data.

Value: Fixed value = "A3D" (for example, 0x41334420)

= MCode (2 bytes)

Meaning: MAKER CODE

Function: Identifies the maker and model of the recorder

Value: High-order 10 bits (maker code); low-order 6 bits (model code)

= CONNUMO (4 bytes)

Meaning: Cumulated number of initially created contents

Function: Designates a unique ID for contents. Even if the contents are edited, the value of the area CONNUMO is not changed.

Value: Same as the contents cumulation number initial key

= BLOCK SERIAL (4 bytes)

Meaning: Serial number assigned to each track

Function: Starts from 0 and increments by 1. Even if the contents are edited, the value of the area BLOCK SERIAL is not changed.

Value: 0 to 0xFFFFFFFF

= BLOCK-SEED (8 bytes)

Meaning: Key for encrypting one block

Function: The beginning of the block is a random number generated by the security block of the recorder. The random number is followed by a value incremented by 1. When the value of the area BLOCK-SEED is lost, since sound is not generated for around one second equivalent to one block, the same data is written to the header and the end of the block. Even if the contents are edited, the value of the area BLOCK-SEED is not changed.

Value: Initially 8-bit random number

= INITIALIZATION VECTOR (8 bytes)

Meaning: Value necessary for encrypting/decrypting ATRAC3 data

Function: Represents an initial value necessary for encrypting and decrypting ATRAC3 data for each block. A block starts from 0. The next block starts from the last encrypted 8-bit value at the last SU. When a block is divided, the last eight bytes just before the start SU is used. Even if the contents are edited, the value of the area INITIALIZATION VECTOR is not changed.

Value: 0 to 0xFFFFFFFFFFFFFFFF

= SU-nnn

Meaning: Data of sound unit

Function: Represents data compressed from 1024 samples. The number of bytes of output data depends on the compression mode. Even if the contents are edited, the value of the area SU-nnn is not changed. For example, in the SP mode, N = 384 bytes.

Value: Data value of ATRAC3

[0114]

In Fig. 18, since N = 384, 42 SU are written to one block. The first two slots (4 bytes) of one block are used as a header. In the last slot (two bytes), the areas BLKID-A3D, MCode, CONNUM0, and BLOCK SERIAL are redundantly written. Thus, M bytes of the remaining area of one block is (16,384 - 384 x 42 - 16 x 3 = 208) bytes. As described above, the eight-byte area BLOCK SEED is redundantly recorded.

[0115]

Next, a management file according to a second embodiment of the present invention will be described. Fig. 25 shows the file structure according to the second embodiment of the present invention. Referring to Fig. 25, a music directory contains a track information management file TRKLIST.MSF (hereinafter, referred to as TRKLIST), a backup track information management file TRKLISTB.MSF (hereinafter, referred to as TRKLISTB), an additional information file INFLIST.MSF (that contains an artist name, an ISRC code, a time stamp, a still picture data, and so forth (this file is referred to as INFIST)), an ATRAC3 data file A3Dnnnn.MSF (hereinafter, referred to as A3nnnn). The file TRKLIST contains two areas NAME1 and NAME2. The area NAME1 is an area that contains the memory card name and the program name (for one byte code corresponding to ASCII/8859-1 character code). The area NAME2 is an area that contains the memory card name and the program name (for two byte code corresponding to MS-JIS/Hankul/Chinese code).

[0116]

Fig. 26 shows the relation between the track information management file TRKLIST, the areas NAME1 and NAME2, and the ATRAC3 data file A3Dnnnn. The file TRKLIST is a fixed-length file of 64 kbytes (= 16 k x 4). An area of 32 kbytes of the file is used for managing tracks. The remaining area of 32 kbytes is

used to contain the areas NAME1 and NAME2. Although the areas NAME1 and NAME2 for program names may be provided as a different file as the track information management file, in a system having a small storage capacity, it is convenient to totally manage the track information management file and program name files.

[0117]

The track information area TRKINF-nnnn and part information area PRTINF-nnnn of the track information management file TRKLIST are used to manage the data file A3Dnnnn and the additional information INFLIST. Only the ATRAC3 data file A3Dnnnn is encrypted. In Fig. 26, the data length in the horizontal direction is 16 bytes (0 to F). A hexadecimal number in the vertical direction represents the value at the beginning of the current line.

[0118]

According to the second embodiment, three files that are the track management file TRKLIST (including a program title file), the additional information management file INFLIST, and the data file A3Dnnnn are used. According to the first embodiment (see Figs. 6, 7, and 8), two files that are the reproduction management file PBLIST for managing all the memory card and the data file ATRAC3 for storing programs are used.

[0119]

Next, the data structure according to the second embodiment will be described. For simplicity, in the data structure according to the second embodiment, the description of similar portions to those of the first embodiment is omitted.

[0120]

Fig. 27 shows the detailed structure of the track information management file TRKLIST. In the track information management file TRKLIST, one cluster (block) is composed of 16 kbytes. The size and data of the file TRKLISTB are the same as those of the backup file TRKLISTB. The first 32 bytes of the track information management file are used as a header. As with the header of the reproduction management file PBLIST, the header of the file TRKLIST contains a BLKID-TL0/TL1 (backup file ID) area (4 bytes), an area T-TRK (2 bytes) for the number of total tracks, a maker code area MCode (2 bytes), an area REVISION (4 bytes) for the number of TRKLIST rewrite times, and an area S-YMDhms (4 bytes) (option) for update date and time data. The meanings and functions of these data areas are the same as those of the first embodiment. In addition, the file TRKLIST contains the following areas.

[0121]

= YMDhms (4 bytes)

Represents the last update date (year, month,

day) of the file TRKLIST.

= N1 (1 byte) (Option)

Represents the sequential number of the memory card (numerator side). When one memory card is used, the value of the area N1 is 0x01.

= N2 (1 byte) (Option)

Represents the sequential number of the memory card (denominator side). When one memory card is used, the value of the area N2 is 0x01.

= MSID (2 bytes) (Option)

Represents the ID of a memory card. When a plurality of memory cards is used, the value of the area MSID of each memory card is the same (T.B.D.). (T.B.D. (to be defined) represents that this value may be defined in future).

= S-TRK (2 bytes).

Represents a special track (T.B.D.). Normally, the value of the area S-TRK is 0x0000.

= PASS (2 bytes) (Option)

Represents a password (T.B.D.).

= APP (2 bytes) (Option)

Represents the definition of a reproduction application (T.B.D.) (normally, the value of the area APP is 0x0000).

= INF-S (2 bytes) (Option)

Represents the additional information pointer of the entire memory card. When there is no additional

information, the value of the area INF-S is 0x00.

[0122]

The last 16 bytes of the file TRKLIST are used for an area BLKID-TL0, an area MCode, and an area REVISION that are the same as those of the header. The backup file TRKLISTB contains the above-described header. In this case, the header contains an area BLKID-TL1, an area MCode, and an area REVISION.

[0123]

The header is followed by a track information area TRKINF for information with respect to each track and a part information area PRTINF for information with respect to each part of tracks (music programs). Fig. 27 shows the areas preceded by the area TRKLIST. The lower portion of the area TRKLISTB shows the detailed structure of these areas. In Fig. 27, a hatched area represents an unused area.

[0124]

The track information area TRKINF-nnn and the part information area PRTINF-nnn contain areas of an ATRAC3 data file. In other words, the track information area TRKINF-nnn and the part information area PRTINF-nnn each contain a reproduction restriction flag area LT (1 byte), a contents key area CONTENTS KEY (8 bytes), a recorder security block serial number area MG(D) SERIAL (16 bytes), an area XT (2 bytes) (option) for representing a feature portion of a music program,

an area INX (2 bytes) (option), an area YMDhms-S (4 bytes) (option), an area YMDhms-E (4 bytes) (option), an area MT (1 byte) (option), an area CT (1 byte) (option), an area CC (1 byte) (option), an area CN (1 byte) (option) (these areas YMDhms-S, YMDhms-E, MT, CT, CC, and CN are used for reproduction restriction information and copy control information), an area A (1 byte) for part attribute, a part size area PRTSIZE (4 bytes), a part key area PRTKEY (8 bytes), and a contents cumulation number area CONNUM (4 bytes). The meanings, functions, and values of these areas are the same as those of the first embodiment. In addition, the track information area TRKINF-nnn and the part information area PRTINF-nnn each contain the following areas.

[0125]

= T0 (1 byte)

Fixed value (T0 = 0x74)

= INF-nnn (Option) (2 bytes)

Represents the additional information pointer (0 to 409) of each track. 00: music program without additional information.

= FNM-nnn (4 bytes)

Represents the file number (0x0000 to 0xFFFF) of an ATRK3 data file.

The number nnnn (in ASCII) of the ATRAC3 data file name (A3Dnnnn) is converted into 0xnnnnn.

= APP_CTL (4 bytes) (Option)

Represents an application parameter (T.B.D.) (Normally, the value of the area APP_CTL is 0x0000).

= P-nnn (2 bytes)

Represents the number of parts (1 to 2039) that compose a music program. This area corresponds to the above-described area T-PART.

= PR (1 byte)

Fixed value (PR = 0 x 50).

[0126]

Next, the areas NAME1 (for one byte code) and NAME2 (for two byte code) for managing names will be described. Fig. 28 shows the detailed structure of the area NAME1 (for one byte code area). Each of the areas NAME1 and NAME2 (that will be described later) is segmented with eight bytes. Thus, their one slot is composed of eight bytes. At 0x8000 that is the beginning of each of these areas, a header is placed. The header is followed by a pointer and a name. The last slot of the area NAME1 contains the same areas as the header.

[0127]

= BLKID-NM1 (4 bytes)

Represents the contents of a block (fixed value) (NM1 = 0x4E4D2D31).

= PNM1-nnn (4 bytes) (Option)

Represents the pointer to the area NM1 (for

one byte code).

= PNM1-S

Represents the pointer to a name representing a memory card.

nnn (= 1 to 408) represents the pointer to a music program title.

The pointer represents the start position (2 bytes) of the block, the character code type (2 bits), and the data size (14 bits).

= NM1-nnn (Option)

Represents the memory card name and music program title for one byte code (variable length). An end code (0x00) is written at the end of the area.

[0128]

Fig. 29 shows the detailed data structure of the area NAME2 (for two byte code). At 0x8000 that is the beginning of the area, a header is placed. The header is followed by a pointer and a name. The last slot of the area NAME2 contains the same areas as the header.

[0129]

= BLKID-NM2 (4 bytes)

Represents the contents of a block (fixed value) (NM2 = 0x4E4D2D32).

= PNM2-nnn (4 bytes) (Option)

Represents the pointer to the area NM2 (for two byte code).

PNM2-S represents the pointer to the name representing the memory card. nnn (= 1 to 408) represents the pointer to a music program title.

The pointer represents the start position (2 bytes) of the block, the character code type (2 bits), and the data size (14 bits).

= NM2-nnn (Option)

Represents the memory card name and music program title for two byte code (variable). An end code (0x0000) is written at the end of the area.

[0130]

Fig. 30 shows the data arrangement (for one block) of the ATRAC3 data file A3Dnnnn in the case that 1 SU is composed of N bytes. In this file, one slot is composed of eight bytes. Fig. 30 shows the values of the top portion (0x0000 to 0x3FF8) of each slot. The first four slots of the file are used for a header. As with the data block preceded by the attribute header of the data file (see Fig. 18) of the first example, a header is placed. The header contains an area BLKID-A3D (4 bytes), a maker code area MCode (2 bytes), an area BLOCK SEED (8 bytes) necessary for encrypting process, an area CONNUM0 (4 bytes) for the initial contents cumulation number, a serial number area BLOCK SERIAL (4 bytes) for each track, and an area INITIALIZATION VECTOR (8 bytes) necessary for encrypting/decrypting process. The second last slot of

the block redundantly contains an area BLOCK SEED. The last slot contains areas BLKID-A3D and MCode. As with the first embodiment, the header is followed by the sound unit data SU-nnnn.

[0131]

Fig. 31 shows the detailed data structure of the additional information management file INFLIST that contains additional information. In the second embodiment, at the beginning (0x0000) of the file INFLIST, the following header is placed. The header is followed by the following pointer and areas.

[0132]

= BLKID-INF (4 bytes)

Represents the contents of the block (fixed value) (INF = 0x494E464F).

= T-DAT (2 blocks)

Represents the number of total data areas (0 to 409).

= MCode (2 bytes)

Represents the maker code of the recorder

= YMDhms (4 bytes)

Represents the record updated date and time.

= INF-nnnn (4 bytes)

Represents the pointer to the area DATA of the additional information (variable length, as 2 bytes (slot) at a time). The start position is represented

with the high order 16 bits (0000 to FFFF).

= DataSlot-0000 (0x0800)

Represents the offset value from the beginning (as a slot at a time).

The data size is represented with low order 16 bits (0001 to 7FFF). A disable flag is set at the most significant bit. MSB = 0 (Enable), MSB = 1 (Disable)

The data size represents the total data amount of the music program.

(The data starts from the beginning of each slot. (The non-data area of the slot is filled with 00.)

The first INF represents a pointer to additional information of the entire album (normally, INF-409).

[0133]

Fig. 32 shows the structure of additional information. An 8-byte header is placed at the beginning of one additional information data area. The structure of the additional information is the same as that of the first embodiment (see Fig. 13C). In other words, the additional information contains an area IN (2 bytes) as an ID, an area key code ID (1 byte), an area SIZE (2 bytes) that represents the size of each additional information area, and a maker code area MCode (2 bytes). In addition, the additional

information contains an area SID (1 byte) as a sub ID.

[0134]

According to the second embodiment of the present invention, in addition to the file system defined as a format of the memory card, the track information management file TRKLIST for or music data is used. Thus, even if the FAT is destroyed, the file can be recovered. Fig. 33 shows a flow of a file recovering process. To recover the file, a computer that operates with a file recovery program and that can access the memory card and a storing device (hard disk, RAM, or the like) connected to the computer are used. The computer has a function equivalent to the DSP30. Next, a file recovering process using the track management file TRKLIST will be described.

[0135]

All blocks of the flash memory whose FAT has been destroyed are searched for TL-0 as the value (BLKID) at the top position of each block. In addition, all the blocks are searched for NM-1 as the value (BLKID) at the top position of each block. Thereafter, all the blocks are searched for NM-2 as the value (BLKID) at the top position of each block. All the contents of the four blocks (track information management file) are stored to for example a hard disk by the recovery computer.

[0136]

The number of total tracks is obtained from data after the fourth byte of the track information management file. The 20-th byte of the track information area TRKINF-001, the value of the area CONNUM-001 of the first music program, and the value of the next area P-001 are obtained. The number of parts is obtained with the value of the area P-001. The values of the areas PRTSIZE of all parts of the track 1 of the area PRTINF is obtained. The number of total blocks (clusters) n is calculated and obtained.

[0137]

After the track information management file is obtained, the flow advances to step 102. At step 102, a voice data file (ATRAC3 data file) is searched. All blocks of other than the management file is searched from the flash memory. Blocks whose top value (BLKID) is A3D are collected.

[0138]

A block of which the value of the area CONNUM0 at the 16-th byte of A3Dnnnn is the same as that of the area CONNUM-001 of the first music program of the track information management file and of which the value of the area BLOCK SERIAL that starts from 20-th byte is 0 is searched. After the first block is obtained, a block (cluster) with the same value of the area CONNUM value as the first block and of which the value of BLOCK SERIAL is incremented by 1 (1 = 0 + 1)

is searched. After the second block is obtained, a block with the same value of the area CONNUM0 as the second block and of which the value of the area BLOCK SERIAL is incremented by 1 (2 = 1 + 1) is searched.

[0139]

By repeating the process, the ATRC3 data file is searched until n blocks (clusters) of the track 1 are obtained. When all the blocks (clusters) are obtained, they are successively stored to the hard disk.

[0140]

The same process for the track 1 is performed for the track 2. In other words, a block of which the value of the area CONNUM0 is the same as that of the area CONNUM-002 of the first music program of the track information management file and of which the value of the area BLOCK SERIAL that starts at the 20-th byte is searched. Thereafter, in the same manner as the track 1, the ATRAC3 data file is searched until the last block (cluster) n' is detected. After all blocks (clusters) are obtained, they are successively stored to the hard disk.

[0141]

By repeating the above-described process for all tracks (the number of tracks: m), all the ATRAC3 data is stored to the hard disk controlled by the recovering computer.

[0142]

At step 103, the memory card whose the FAT has been destroyed is re-initialized and then the FAT is reconstructed. A predetermined directory is formed in the memory card. Thereafter, the track information management file and the ATRAC3 data file for m tracks are copied from the hard disk to the memory card. Thus, the recovery process is finished.

[0143]

In the management file and data file, important parameters (in particular, codes in headers) may be recorded triply rather than doubly. When data is redundantly recorded, the same data may be recorded at any positions as long as they are apart from each other for one page or more.

[0144]

In the first and second embodiments, as an example of the player/recorder of the system audio set, a memory card recorder was described. According to the present invention, a digital signal reproduced by a CD player is stored to a hard disk. The hard disk is used as an audio server. The digital signal is moved from the hard disk to a memory card 40 having the above-described format. Thus, with the above-described digital audio player/recorder or portable player/recorder, the user can listen to the reproduced digital audio data. Next, corresponding to the first

embodiment shown in Figs. 7 to 23 and the second embodiment shown in Figs. 25 to 32, the structure that moves contents data from the hard disk to the memory card will be described in detail.

[0145]

Fig. 34 is a schematic diagram showing a storing apparatus having a hard disk. The storing apparatus is for example a personal computer. In the following description, the storing unit is simply referred to as host or host side. In Fig. 34, reference numeral 201 is a hard disk drive. The hard disk drive 201 is operated under the control of a CPU 202. In association with the CPU 202, an external non-volatile memory (external NVRAM) 203, an operation button portion 204, and a display device 205 are disposed.

[0146]

In addition, an ATRAC3 audio encoder/decoder 206 is disposed. An analog input signal 207 is supplied to an A/D converter 208. The A/D converter 208 converts the analog signal 207 into a digital audio signal. The audio encoder/decoder 206 compresses the digital audio signal that is output from the A/D converter 208 corresponding to ATRAC3. In addition, a digital input signal 210 is supplied from a CD player 209. The digital input signal 210 is supplied to the audio encoder/decoder 206 through a digital input

receiver 211. The audio encoder/decoder 206 compresses the digital input signal 210 that is received from the digital input receiver 211 corresponding to ATRAC3. The host side decodes audio data stored in the hard disk drive 201. The audio encoder/decoder 206 decodes the audio data that is read from the hard disk drive 201 into a digital audio signal. The digital audio signal is supplied to a D/A converter 213. The D/A converter 213 converts the digital audio signal that is received from the audio encoder/decoder 206 into an analog audio signal. The D/A converter 213 outputs an analog audio signal 214. Alternatively, compressed/non-compressed digital audio data may be downloaded to the hard disk HDD 201 through Internet and a public telephone line (not shown).

[0147]

The compressed audio data is supplied from the audio encoder/decoder 206 to a security block S-SAM (D) 212 of the host side. The security block S-SAM (D) 212 encrypts the compressed audio data. As with the audio recorder, the compressed audio data is encrypted using a contents key. The encrypted ATRAC3 data is stored to the hard disk drive 201 under the control of the CPU 202. In the case of the digital input signal, information such as ISRC (Industry Standard Recoding Code) and TOC (Table Of Content)_ID that identify music programs recorded on a disc can be obtained. The

security block S-SAM (D) 212 generates a contents key and a contents cumulation number CONNUM for each contents title (audio file (track) in the first embodiment). In addition, each host is assigned a unique serial number. These values are stored in the hard disk drive 201 and/or the external non-volatile memory 203.

[0148]

To allow an encrypted ATRAC3 data file stored in the hard disk drive 201 to be reproduced by other than the unit (host) that has encrypted the ATRAC3 data file, the encrypted ATRAC3 data file is moved to the memory card 40. The moved data file is not left in the hard disk unlike with the copying process.

[0149]

Since the ATRAC3 data has been encrypted with a contents key, unless it is decrypted on the copied side, it cannot be reproduced. However, when the contents key as an encrypting key is stolen, encrypted data can be easily decrypted. To prevent such a problem, the contents key itself is encrypted. The contents key is not exposed to the outside. For example, when ATRAC3 data is moved from the hard disk drive 201 to the memory card 40, the contents key is encrypted with a session key. The encrypted contents key is sent from the hard disk dive 201 to the memory card 40. The memory card 40 decrypts the contents key

with the session key. Thereafter, the memory card 40 encrypts the contents data with a storage key thereof. The encrypted contents key is stored in the memory card 40.

[0150]

Likewise, when data is moved from the memory card 40 to the hard disk drive 201, the memory card 40 encrypts a contents key with a session key and sends the encrypted contents key to the hard disk drive 201. Thus, the value of the contents key stored in the hard disk drive 201 is different from the value of the contents key stored in the memory card 40. Thus, a pair of audio data and contents key should be stored on the moved side.

[0151]

Next, with reference to Fig. 35, the data move process will be described in detail. First of all, a data move process for moving data formatted for the audio player/recorder shown in Fig. 1 and recorded in the memory card 40 to the hard disk drive 201 of the host side will be described. In the initial state of which the power of the host side is turned on, it is determined whether or not the memory card 40 has been attached. When the memory card 40 has been attached, the host side and the memory card 40 are authenticated each other. When they have been successfully authenticated, the host side and the memory card side

share a session key Sek.

[0152]

Next, the host reads data from the memory card 40. According to the first embodiment of the present invention, the contents key CK is read from the reproduction management file PBLIST. In contrast, according to the second embodiment of the present invention, a contents key CK (DES (Data Encryption Standard) (Kstm, CK)) encrypted with a storage key Kstm that is unique to each memory card 40 is extracted from the track information area TRKINF. The DES (Kstm, CK) is sent from the host to the memory card 40. The memory card 40 decrypts the encrypted contents key DES (Kstm, CK) with the storage key Kstm. The decrypted contents key is encrypted with the session key Sek.

[0153]

The contents key DES (Sek, CK) encrypted with the session key Sek is sent from the memory card 40 to the host side. The host side decrypts the contents key CK with the session key Sek, re-encrypts the decrypted contents key CK with a storage key Kstd that is unique thereto, and stores the re-encrypted storage key to the hard disk drive 201. In other words, the key is stored as a new contents key. The storage keys Kstd and Kstm are stored in such a manner that their values cannot be read from the outside.

[0154]

In Fig. 35, a security block 212a of the host side and a security block of the memory card 40 authenticate each other and they share a session key Sek. The security block 212a supplies a storage key Kstd and a contents key CK to an encrypting device 212b. The encrypting device 212b creates an encrypted contents key DES (Dstd, CK).

[0155]

As denoted by a path 215, encrypted ATRAC3 data is moved from the memory card 40 to the host side. The ATRAC3 data is stored to the hard disk drive 201. In this case, as described with reference to Fig. 27, the track management information TRKINF recorded in the memory card 40 is sent to the host side along with a data file. In particular, the contents cumulation number (CONNUM), the S-SAM serial number, and the file number FNM-nnnn for each music program are directly copied to the track information area TRKINF-nnnn and recorded as a track information area TRKINF of the host side. Unlike with the contents key, these attribute information is not encrypted.

[0156]

Unless these information is moved to the host side, even if audio data is stored to the hard disk drive 20, the audio data stored in the host cannot be decrypted. Unless the audio data stored in the hard disk is moved to the memory card, the audio data cannot

be reproduced.

[0157]

The contents cumulation number CONNUM is a cumulation number of which each music program is recorded through encrypting devices of security blocks of the memory card 40 and the host side. The contents cumulation number CONNUM has a combination of 2^{32} = 4,200,000,000. The non-volatile memory of each encrypting device stores the last contents cumulation number. Thus, the contents cumulation number is not redundant in each memory card. The S-SAM serial number (SERIAL) is a number unique to each encrypting device. The S-SAM serial number has a combination of 2^{128}. Thus, the S-SAM serial number is not redundant. The file number FNM-nnnn is a number assigned to each ATRAC3 data file. The file number FNM-nnnn is assigned by hardware. Thus, the file number FNM-nnnn may be redundant. Consequently, the contents cumulation number CONNUM and the S-SAM serial number (SERIAL) are added as auxiliary numbers. Thus, with a total of three types of numbers, a data file is recorded, the file can be uniquely existed.

[0158]

As described above, to perform an authenticating process and an encrypting process, the security block 212 of the host side creates or provides:

self unique number (S-SAM serial number),

contents key CK (created for each contents title),

storage key Kstd, and

session key Sek

[0159]

According to the second embodiment of the present invention, the hard disk drive 201 of the host side and/or the external non-volatile memory 203 has a track information area TRKINF paired with an audio data file. The track information area TRKINF contains:

file number FNM-nnnn,

encrypted contents key CK,

S-SAM serial number, and

contents cumulation number CONNUM.

[0160]

When digital data is directly recorded from for example the CD player 209 to the hard disk drive 201, the audio encoder/decoder 206 compresses audio data corresponding to ATRAC3. The security block 212 of the host side creates a contents key CK for each contents title (music program) and encrypts the contents key with the storage key Kstd unique thereto. The encrypting device 212c encrypts ATRAC3 data with the encrypted contents key DES (Kstd, CK) and stores the encrypted audio data 216 to the hard disk drive 201. At this point, the security block 212a of the

host side creates the contents cumulation number CONNUM and the S-SAM (D) serial number for each music program. According to the first embodiment of the present invention, the contents cumulation number CONNUM and the S-SAM (D) serial number are stored as the A3Dnnnn.MSA (ATRAC data file) shown in Fig. 17. According to the second embodiment of the present invention, the contents cumulation number CONNUM and the S-SAM (D) serial number are stored as the track information area TRKINF to the hard disk drive 201. However, these attribute information is not encrypted with the storage key Kstd unlike with the contents key.

[0161]

In addition, the host itself decrypts and reproduces contents data stored in the hard disk drive 201. With the operation button portion 204, the user can record and reproduce contents data on the host side with reference to information displayed on the display device 205.

[0162]

When digital data is copied from the CD player 209 to the hard disk drive 201 of the host side, the digital receiver 211 can obtain information that identifies a music program recorded on a CD (the information is for example TOC_ID or ISRC of each music program). When digital data received from the CD player 209 is copied, the digital receiver 211 assigns

a directory name for each CD.

[0163]

In contrast, data can be moved from the host side to the memory card 40. In this case, the host side and the memory card 40 authenticate each other. When they have successfully authenticated each other, they share a session key Sek. The host reads a contents key DES (Kstd, CK) from the hard disk drive 201 and decrypts it with a storage key Kstd. The host encrypts the decrypted contents key with a session key Sek and sends the encrypted contents key DES (Sek, CK) to the memory card 40.

[0164]

The memory card 40 decrypts a contents key CK with a session key Sek. Thereafter, the memory card 40 re-encrypts the contents key CK with a storage key Kstm that is unique thereto. According to the first embodiment of the present invention, the encrypted contents key DES (Kstm, CK) is stored in the reproduction management file PBLIST and the ATRAC data file. According to the second embodiment of the present invention, the encrypted contents key DES (Kstm, CK) is stored in the track information area TRKINF. Information (for example, contents cumulation number CONNUM and S-SAM () serial number) other than the contents key is not re-encrypted, but directly recorded.

[0165]

According to an embodiment of the present invention, to securely prevent audio data from being illegally copied, when the audio data is moved from the hose side to a memory card 40, information that represents a move history is stored to an external non-volatile memory 203. In other words, the hose side manages the move history that represents what music programs have been moved. Since the move history is stored to the external non-volatile memory 203 rather than the hard disk drive HDD 201, audio data recorded in the hard disk drive HDD 201 can be prevented from being illegally copied to a memory card. In other words, unless move information is recorded in the hard disk drive HDD 201, even if audio data recorded therein is illegally copied, moved data cannot be moved again.

[0166]

Fig. 36 shows a process for preventing audio data from being illegally copied. First of all, a copy process for copying audio data from a hard disk drive HDD 1 that stores audio data will be described. Before performing a move process (that will be described later), 10 music programs stored in the hard disk drive HDD 1 are copied to a hard disk drive HDD 2. A host side CPU 202 and an external non-volatile memory 203 manage move history information. Thereafter, as described above, 10 music programs and a contents key

that have been encrypted are moved from the hard disk drive HDD 1 to a first memory card 40X. In this case, as a precondition, the memory card 40X should have been correctly authenticated with the host side. When the audio data is moved, the encrypted contents key necessary for decrypting the audio data that has been moved to the memory card 40X is also sent to the memory card 40X. In such a manner, 10 music programs of audio data are completely moved from the hard disk drive HDD 1 to the memory card 40X.

[0167]

Next, a move process for moving 10 music programs of music data from the hard disk drive HDD 1 to the hard disk drive HDD 2 will be described. In this case, a second memory card 40Y is used. Since the host side has a security block 212, it correctly authenticates the memory card 40Y and shares a session key Sek with the memory card 40Y. Thus, the contents key CK encrypted with the session key Sek can be moved from the hard disk drive HDD 2 to the memory card 40Y. After the memory card 40Y is correctly authenticated, when the encrypted data is moved to the memory card 40Y, the data stored therein can be decrypted and reproduced. When music programs are copied to a plurality of hard disk drives and the music programs are copied from a hard disk drive to a memory card, the music programs can be unlimitedly copied. Thus, the

copyright of the music programs are violated. When 10 music programs stored in the hard disk drive HDD 1 on one host side are copied or moved to a hard disk drive on another host side, move history information stored in an external non-volatile memory NVRAM is prohibited from being copied/moved to the hard disk drive HDD 2.

[0168]

Corresponding to a flow chart shown in Fig. 37, the host side CPU 202 references the history information stored in the non-volatile memory 203 and determines whether or not to permit audio data to be moved. The memory card 40 sends a move request that designates a music program stored in the hard disk drive HDD 201 to the CPU 202 (at step S201). Thereafter, the CPU 202 checks the external non-volatile memory 203 for the move history of the designated music program (at step S202). In other words, the CPU 202 determines whether or not the designated music program has been moved corresponding to the move request (at step S203).

[0169]

When the determined result at step S203 is No, the flow advances to step S204. At step S204, the designated music program is moved from the host side hard disk drive HDD 201 to the memory card 40 (at step S204). In addition, the move history is recorded to the external non-volatile memory 203. When the

determined result at step S203 is Yes, the host side CPU 202 prohibits the designated music program from being moved from the hard disk drive HDD 201 (at step S205). In this case, the display device 205 displays a message that represents that the designated music program has been moved. Alternatively, a synthesizing means may generates an audio message that represents that the designated music program has been moved.

[0170]

In the above description, data communication between a hard disk drive and a memory card that are storing units was described. Alternatively, a host having a hard disk drive (in this case, the host is for example a personal computer) may interface with a terminal unit of an electronic contents delivering system. In this case, a process similar to a move process performed between the hard disk drive and the memory card is performed between the terminal unit and the personal computer.

[0171]

In the above embodiment, the case that contents data is audio data was described. Of course, the present invention can be applied to video data, program data, and so forth other than audio data. In addition, the present invention can be applied to other storage mediums such as a magneto-optical disc, a phase change type disc, and a semiconductor memory other than

a hard disk.

[0172]

[Effect of the Invention]

According to the present invention, an encrypting device is also disposed on the storing unit side. A contents key encrypted with a session key and contents data (data file) encrypted with the contents key are received from a memory card as a storage medium. After the contents key is decrypted with the session key, the contents key is re-encrypted with a key unique to the storing unit. Since the contents key is re-keyed, even if the contents data is moved to other than the original memory card, the contents data can be decrypted. In addition, when contents data is moved from the storing unit to the memory card, the contents key is re-keyed. Thus, the contents moved to a memory card can be decrypted by another unit.

[0173]

In addition to a medium that stores contents data, move history information is stored in a non-volatile memory. Thus, contents data of a medium can be securely prevented from being illegally copied to another medium.

[Brief Description of the Drawings]

[Fig. 1]

Block diagram showing the entire structure of the present invention.

[Fig. 2]

Block diagram showing the structure of a DSP according to the embodiment of the present invention.

[Fig. 3]

Block diagram showing the structure of a memory card according to the embodiment of the present invention.

[Fig. 4]

Schematic diagram showing the file system process hierarchy of the flash memory according to the embodiment of the present invention.

[Fig. 5]

Schematic diagram showing the physical structure of a data in a flash memory according to the embodiment of the present invention.

[Fig. 6]

Schematic diagram showing a file convention according to the embodiment of the present invention.

[Fig. 7]

Schematic diagram showing the data structure of the reproduction management file according to the embodiment of the present invention.

[Fig. 8]

Schematic diagram showing the data structure of the data file according to the embodiment of the present invention.

[Fig. 9]

Schematic diagram showing the structure of a data file according to the embodiment of the present invention.

[Fig. 10]

Schematic diagram showing an example of the editing process of the data file according to the embodiment of the present invention.

[Fig. 11]

Schematic diagram showing the other example of the editing process of the data file according to the embodiment of the present invention.

[Fig. 12]

Schematic diagram showing the structure of a reproduction management file according to the embodiment of the present invention.

[Fig. 13]

Schematic diagram showing the structure of an additional information area and the part of a reproduction management file.

[Fig. 14]

Schematic diagram showing an example of an additional information according to the embodiment of the present invention.

[Fig. 15]

Schematic diagram showing an example of an additional information according to the embodiment of the present invention.

[Fig. 16]

Schematic diagram showing an example of an additional information according to the embodiment of the present invention.

[Fig. 17]

Schematic diagram showing a practical data structure of an additional information according to the embodiment of the present invention.

[Fig. 18]

Schematic diagram showing the structure of the data file according to the embodiment of the present invention.

[Fig. 19]

Schematic diagram showing the part of an attribute header of the data file.

[Fig. 20]

Schematic diagram showing the part of an attribute header of the data file.

[Fig. 21]

Schematic diagram showing the types of recording mode and recording hours and or the like for each recording mode according to the embodiment of the present invention.

[Fig. 22]

Schematic diagram explaining the copy control information according to the embodiment of the present invention.

[Fig. 23]

Schematic diagram showing the part of an attribute header in the data file.

[Fig. 24]

Schematic diagram showing the header of each data block in the data file.

[Fig. 25]

Schematic diagram showing the convention of the other data structure of the file which can be used for this invention.

[Fig. 26]

Schematic diagram showing the relation between files of the other data structure.

[Fig. 27]

Schematic diagram showing the structure of the the track management file of the other data structure.

[Fig. 28]

Schematic diagram showing the structure of the name file in the track information management file of the other data structure.

[Fig. 29]

Schematic diagram showing the structure of the name file in the track information management file of the other data structure.

[Fig. 30]

Schematic diagram showing the structure of

the data file of the other data structure.

[Fig. 31]

Schematic diagram showing the structure of the additional information management file of the other data structure.

[Fig. 32]

Schematic diagram showing the structure of the additional information data of the other data structure.

[Fig. 33]

Schematic diagram explaining the flow of the file recovery process.

[Fig. 34]

Block diagram explaining a move process according to the embodiment of the present invention.

[Fig. 35]

Schematic diagram explaining a re-keying operation in the move process.

[Fig. 36]

Schematic diagram explaining the prevention of the illegal copy by the copy of the hard disk.

[Fig. 37]

Flow chart explaining the prevention of the illegal copy by the copy of the hard disk.

[Description of Reference Numerals]

12 ... Audio decoder, 30 ... DSP, 40 ... Memory card, TRKLIST.MSF ... Track information management file,

INFLIST.MSF ... Additional information management file, A3Dnnn.MSA ... Audio data file, 201 ... Hard disk drive, 202 ... CPU, 203 ... External non-volatile memory, 206 ... Audio encoder/decoder, 212 ... Security block, CK ... Contents key, Sek ... Session key, Kstm ... Unique storage key for the memory card, Kstd ... Unique storage key for storing apparatus

Title of Document] Abstract

[Abstract]

[Subject]

The move of contents is performed smoothly with high security.

[Solving means]

The host has an attachable/detachable memory card 40 and the hard disk drive 201, shares the session key, when they have been successfully authenticated. contents key CK is encrypted by a unique storage key of the memory card that resides on the memory card 40. Contents key CK is decrypted that is encrypted by the session key Sek and transmitted to the host side. Encrypted data file and file attribute information are transmitted directly to the host side via a route 215. In the host side, contents key CK is decrypted by the session key Sek that is encrypted by a unique storage key Kstd of the host is allocated. Likewise contents are moved from the host to the memory card 40, the content's key is re-keyed. A move history information is managed on the non-volatile memory of the host side.

[Selected Drawing] Fig. 35

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT(S): Nobuyuki KIHARA and Teppei YOKOTA

SERIAL No.: 09/674,441 Group Art Unit: 2143

FILED: November 1, 2000 Examiner: Kyung H Shin

INVENTION: DATA PROCESSING APPARATUS, DATA PROCESSING METHOD, TERMINAL UNIT, AND TRANSMISSION METHOD OF DATA PROCESSING APPARATUS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

CERTIFIED TRANSLATION

Yuka NAKAMURA residing at c/o SUGIURA PATENT OFFICE, 7th floor, Ikebukuro Park Bldg., 49-7, Minami Ikebukuro 2-chome, Toshima-ku, Tokyo, JAPAN, declares:

(1) that she knows well both the Japanese and English languages;

(2) that she translated Japanese Application No. 11-178188 from Japanese to English;

(3) that the attached English translation is a true and correct translation of the above-identified Japanese Application to the best of her knowledge and belief; and

(4) that all statements made of her own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements are made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 USC 1001, and that such false statements may jeopardize the validity of the application or any patent issuing thereon.

November 17, 2004 Yuka Nakamura

Date Yuka NAKAMURA

[Title of Document] Specification

[Title of the Invention] Non-volatile Memory, Data Processing Apparatus, and Method

[Scope of Claims for a Patent]

[Claim 1]

A non-volatile memory from which a computer can read data and that is attachable to and detachable from the computer,

wherein the non-volatile memory stores:

data file,

first file management information for managing the data file, and

second file management information, used in a file management system of the computer, for managing the data file and the first file management information, and

wherein the first file management information has a fixed length whose data amount is any integer times the data amount of a predetermined data unit.

[Claim 2]

The non-volatile memory as set forth in claim 1,

wherein the first file management information has a header containing an identification code that represents the first file management information.

[Claim 3]

The non-volatile memory as set forth in claim 2,

wherein the identification code is repeatedly placed at positions apart from the header of the first file management information.

[Claim 4]

The non-volatile memory as set forth in claim 1,

wherein the data file managed corresponding to the first file management information contains a contents cumulation number and a sequence number at predetermined positions of each data unit.

[Claim 5]

The non-volatile memory as set forth in claim 1,

wherein the first file management information contains revision information at a predetermined position of each data unit, the revision information being varied whenever the data file is recorded.

[Claim 6]

The non-volatile memory as set forth in claim 1,

wherein the data file is composed of at least one part, and

wherein the first file management information contains management information for each data file and management information of each part of each data file.

[Claim 7]

The non-volatile memory as set forth in claim 1,

wherein the non-volatile memory further stores:

third file management information managed corresponding to the second file management information,

wherein the third file management information is added as a header at intervals of a predetermined data unit of the data file.

[Claim 8]

The non-volatile memory as set forth in claim 7,

wherein the first file management information contains information that represents the reproduction order of the data file recorded in the non-volatile memory.

[Claim 9]

The non-volatile memory as set forth in claim 7,

wherein the first file management information contains management information of the overall non-volatile memory, and

wherein the third file management information contains management information of each data file.

[Claim 10]

The non-volatile memory as set forth in claim 8,

wherein the first file management information contains additional information of the overall non-volatile memory, and

wherein the third file management information contains additional information of each data file.

[Claim 11]

A data processing apparatus having an attachable/detachable non-volatile memory from which a computer can read data, the apparatus comprising:

a controlling portion for handling a data file, first file management information for managing the data file, and second file management information, used in a file management system, for managing the data file and the first file management information; and

a memory interface disposed between said controlling portion and the non-volatile memory,

wherein the data file and the first file management information are stored from said controlling portion to the non-volatile memory through said memory interface,

wherein the data file and the first file management information are read from the non-volatile memory to said controlling portion through said memory interface, and

wherein the first file management information

has a fixed length whose data amount is any integer times the data amount of a predetermined data unit of the non-volatile memory.

[Claim 12]

The data processing apparatus as set forth in claim 11,

wherein the first file management information has a header containing an identification code that represents the first file management information, and

wherein the first file management information is extracted from the non-volatile memory with reference to the identification code.

[Claim 13]

The data processing apparatus as set forth in claim 12,

wherein the identification code having the same value is repeatedly placed at positions apart from the header of the first file management information.

[Claim 14]

The data processing apparatus as set forth in claim 11,

wherein the data file managed corresponding to the first file management information contains a contents cumulation number and a sequence number at predetermined positions of each data unit.

[Claim 15]

The data processing apparatus as set forth in

claim 11,

wherein the first file management information contains revision information at a predetermined position of each data unit, the revision information being varied whenever the data file is recorded.

[Claim 16]

The data processing apparatus as set forth in claim 15,

wherein the first file management information contains the revision information at the beginning portion and the end portion of the first file management information,

wherein it is determined whether or not the revision information at the beginning portion of the first file management information matches the revision information at the end portion of the first file management information, and

wherein when the revision information at the beginning portion does not match the revision information at the end portion, it is determined that the first file management information has not been correctly rewritten.

[Claim 17]

The data processing apparatus as set forth in claim 11,

wherein each data file is composed of at least one part, and

wherein the first file management information contains management information for each data file and management information of each part of each data file.

[Claim 18]

The data processing apparatus as set forth in claim 11,

wherein the non-volatile memory further stores:

third file management information managed corresponding to the second file management information,

wherein the third file management information is added as a header at intervals of a predetermined data unit of the data file.

[Claim 19]

The data processing apparatus as set forth in claim 18,

wherein the first file management information contains information that represents the reproduction order of the data file recorded in the non-volatile memory.

[Claim 20]

The data processing apparatus as set forth in claim 18,

wherein the first file management information contains management information of the overall non-volatile memory, and

wherein the third file management information contains management information of each data file.

[Claim 21]

The data processing apparatus as set forth in claim 20,

wherein the first file management information contains additional information of the overall non-volatile memory, and

wherein the third file management information contains additional information of each data file.

[Claim 22]

A data processing method for a data processing apparatus from which a computer can read data and that is attachable to and detachable from the computer,

wherein the non-volatile memory stores:

a data file,

first file management information for managing the data file, and

second file management information, used in a file management system, for managing the data file and the first file management information, and

wherein when the second file management information is destroyed, a file is restored with reference to the first file management information.

[Claim 23]

The data processing method as set forth in

claim 22, wherein the file is restored by the steps of:

searching and collecting the first file management information from the non-volatile memory;

searching and collecting the data file; and

initializing the non-volatile memory, re-structuring the second file management information, and storing the collected first file management information and the collected data file to the initialized non-volatile memory.

[Detailed Description of the Invention]

[0001]

[Industrial Field of Utilization]

The present invention relates to a non-volatile memory, a data processing apparatus, and a method thereof in which attachable/detachable memory card is used as a record media for audio data or so.

[0002]

[Prior Art]

EEPROM (Electrically Erasable Programmable ROM) that is an electrical rewritable memory requires a large space because each bit is composed of two transistors. Thus, the integration of EEPROM is restricted. To solve this problem, a flash memory that allows one bit to be accomplished with one transistor using all-bit-erase system has been developed. The flash memory is being expected as a successor of conventional record mediums such as magnetic disks and

optical discs.

[0003]

A memory card using a flash memory is also known. The memory card can be freely attached to an apparatus and detached therefrom. A digital audio recording/reproducing apparatus that uses a memory card instead of a conventional CD (Compact Disc: Trademark) or MD (Mini Disc: Trademark) can be accomplished.

[0004]

A file management system used for a conventional personal computer is named FAT (File Allocation Table). In the FAT system, when a particular file is defined, predetermined parameters are successively set to the file. Thus, the size of a file becomes variable. One file is composed of at least one management unit (sector, cluster, or the like). Data corresponding to the management unit is written to a table referred to as FAT. In the FAT file system, a file structure can be easily formed regardless of the physical characteristics of a record medium. Thus, the FAT file system can be used for a magneto-optical disc as well as a floppy disk and a hard disk. In the above-mentioned memory card, the FAT file system is used.

[0005]

[Problem to be Solved by the Invention]

In the conventional FAT file system, once the

FAT is destroyed, it cannot be almost recovered. Thus, as possible countermeasures, it is necessary to backup data to another medium. Among users of personal computers, such countermeasures are essential. Thus, the users should backup data as their responsibilities. However, it is troublesome for the users to backup data. In addition, to do that, another medium is required.

[0006]

Therefore, an object of the present invention is to provide a non-volatile memory, a data Processing apparatus, and a method thereof non-volatile memory, that allow a file to be recovered even if a file management table is destroyed without need to make a backup file.

[0007]

[Means for Solving the Problem]

According to the invention disclosed in claim 1, a non-volatile memory from which a computer can read data and that is attachable to and detachable from the computer,

wherein the non-volatile memory stores:

data file,

first file management information for managing the data file, and

second file management information, used in a file management system of the computer, for managing

the data file and the first file management information, and

wherein the first file management information has a fixed length whose data amount is any integer times the data amount of a predetermined data unit.

[0008]

According to the invention disclosed in claim 11, a data processing apparatus having an attachable/detachable non-volatile memory from which a computer can read data, the apparatus comprising:

a controlling portion for handling a data file, first file management information for managing the data file, and second file management information, used in a file management system, for managing the data file and the first file management information; and

a memory interface disposed between said controlling portion and the non-volatile memory,

wherein the data file and the first file management information are stored from said controlling portion to the non-volatile memory through said memory interface,

wherein the data file and the first file management information are read from the non-volatile memory to said controlling portion through said memory interface, and

wherein the first file management information has a fixed length whose data amount is any integer

times the data amount of a predetermined data unit of the non-volatile memory.

[0009]

According to the invention disclosed in claim 22, a data processing method for a data processing apparatus from which a computer can read data and that is attachable to and detachable from the computer,

wherein the non-volatile memory stores:

a data file,

first file management information for managing the data file, and

second file management information, used in a file management system, for managing the data file and the first file management information, and

wherein when the second file management information is destroyed, a file is restored with reference to the first file management information.

[0010]

According to the present invention, the first file management information is recorded on the record media, such as nonvolatile memory. Thus, in the case that the second file management information is destroyed, the file can be recovered corresponding to the first management file.

[0011]

[Embodiment of the Invention]

Next, an embodiment of the present invention

will be described. Fig. 1 is a block diagram showing the structure of a digital audio recorder (that has functions as a player and as a recorder) using a memory card according to an embodiment of the present invention. The digital audio recorder records and reproduces a digital audio signal using a attachable/detachable memory card. In reality, the recorder/player composes an audio system along with an amplifying unit, a speaker, a CD player, an MD recorder, a tuner, and so forth. However, it should be noted that the present invention can be applied to other audio recorders. For example, the present invention can be applied to a portable recording apparatus. In addition, the present invention can be applied to a recorder that records a digital audio data that is circulated as a satellite data communication, a digital broadcast, or Internet. Moreover, the present invention can be applied to a system that records/reproduces moving picture data and still picture data rather than audio data. The system according to the embodiment of the present invention can record and reproduce additional information such as picture and text other than a digital audio signal.

[0012]

The recorder has an audio encoder/decoder IC 10, a security IC 20, a DSP (Digital Signal Processor) 30. Each of these devices is composed of a one-chip

IC. 40 is a attachable/detachable memory card to recorder. The one-chip IC of the memory card 40 has flash memory (nonvolatile memory), a memory control block, and a security block. The security block has a DES (Data Encryption Standard) encrypting circuit. According to the embodiment, the recording/reproducing apparatus may use a microcomputer instead of the DSP 30.

[0013]

The audio encoder/decoder IC 10 has an audio interface 11 and an encoder/decoder block 12. The encoder/decoder block 12 encodes a digital audio data corresponding to a highly efficient encoding method and writes the encoded data to the memory card 40. In addition, the encoder/decoder block 12 decodes encoded data that is read from the memory card 40. As the highly efficient encoding method, the ATRAC3 format that is a modification of the ATRAC (Adaptive Transform Acoustic Coding) format used in Mini-Disc is used.

[0014]

In the ATRAC3 format, audio data sampled at 44.1 kHz and quantized with 16 bits is processed. In the ATRAC3 format, the minimum data unit of audio data that is processed is a sound unit (SU). 1 SU is data of which data of 1024 samples (1024 x 16 bits x 2 channels) is compressed to data of several hundred bytes. The duration of 1 SU is around 23 msec. In the

ATRAC 3 , the data amount of audio data is compressed to data that is around 10 times smaller than that of original data. As with the Mini-Disc, the audio signal compressed and decompressed corresponding to the ATRAC3 format less deteriorates in the audio quality.

[0015]

A line input selector 13 selectively supplies the reproduction output signal of an MD, the output signal of a tuner, or a reproduction output signal of a tape to an A/D converter 14. The A/D converter 14 converts the selected line input signal to a digital audio signal (sampling frequency = 44.1 kHz; the number of quantizing bits = 16). A digital input selector 16 selectively supplies a digital output signal of an MD, a CD, or a CS (Satellite Digital Broadcast) to a digital input receiver 17. The digital input signal is transmitted through for example an optical cable. An output signal of the digital input receiver 17 is supplied to a sampling rate converter 15. The sampling rate converter 15 converts the digital input signal into a digital audio signal (sampling frequency = 44.1 kHz).

[0016]

The encoder/decoder block 12 of the audio encoder/decoder IC 10 supplies encoded data to a DES encrypting circuit 22 through an interface 21 of the security IC 20. The DES encrypting circuit 22 has a

FIFO 23. The DES encrypting circuit 22 is disposed so as to protect the copyright of contents. The memory card 40 also has a DES encrypting circuit. The DES encrypting circuit 22 of the recording/reproducing apparatus has a plurality of master keys and an apparatus-unique storage key. The DES encrypting circuit 22 also has a random number generating circuit. The DES encrypting circuit 22 can share an authenticating process and a session key with the memory card 40 that has the DES encrypting circuit. In addition, the DES encrypting circuit 22 can re-encrypt data with the storage key of the DES encrypting circuit.

[0017]

The encrypted audio data that is output from the DES encrypting circuit 22 is supplied to a DSP (Digital Signal Processor) 30. The DSP 30 communicates with the memory card 40 through an interface. In this example, the memory card 40 is attached to an attaching/detaching mechanism (not shown) of the recording/reproducing apparatus. The DSP 30 writes the encrypted data to the flash memory of the memory card 40. The encrypted data is serially transmitted between the DSP 30 and the memory card 40. In addition, an external SRAM (Static Random Access Memory) 31 is connected to the DSP 30. The SRAM 31 provides the recording/reproducing apparatus with a sufficient

storage capacity so as to control the memory card 40.

[0018]

A bus interface 32 is connected to the DSP 30. Data is supplied from an external controller (not shown) to the DSP 30 through a bus 33. The external controller controls all operations of the audio system. The external controller supplies data such as a record command or a reproduction command that is generated corresponding to a user's operation through an operation portion to the DSP 30 through the bus interface 32. In addition, the external controller supplies additional information such as image information and character information to the DSP 30 through the bus interface 32. The bus 33 is a bidirectional communication path. Additional information that is read from the memory card 40 is supplied to the external controller through the DSP 30, the bus interface 32, and the bus 33. In reality, the external controller is disposed in for example an amplifying unit of the audio system. In addition, the external controller causes a display portion to display additional information, the operation state of the recorder, and so forth. The display portion is shared by the audio system. Since data that is exchanged through the bus 33 is not copyright protected data, it is not encrypted.

[0019]

The encrypted audio data that is read from the memory card 40 by the DSP 30 is decrypted by the security IC 20. The audio encoder/decoder IC 10 decodes the encoded data corresponding to the ATRAC3 format. Output data of the audio encoder/decoder 10 is supplied to a D/A converter 18. The D/A converter 18 converts the output data of the audio encoder/decoder 10 into an analog signal. The analog audio signal is supplied to a line output terminal 19.

[0020]

The analog audio signal is supplied to an amplifying unit (not shown) through the line output terminal 19. The analog audio signal is reproduced from a speaker or a head set. The external controller supplies a muting signal to the D/A converter 18. When the muting signal represents a mute-on state, the external controller prohibits the audio signal from being output from the line output terminal 19.

[0021]

Fig. 2 is a block diagram showing the internal structure of the DSP 30. Referring to Fig. 2, the DSP 30 comprises a core 34, a flash memory 35, an SRAM 36, a bus interface 37, a memory card interface 38, and inter-bus bridges. The DSP 30 has the same function as a microcomputer. The core 34 is equivalent to a CPU. The flash memory 35 stores a program that causes the DSP 30 to perform predetermined processes.

The SRAM 36 and the external SRAM 31 are used as a RAM of the recording/reproducing apparatus.

[0022]

The DSP 30 controls a writing process for writing encrypted audio data and additional information to the memory card 40 corresponding to an operation signal such as a record command received through the bus interfaces 32 and 37 and a reading process for reading them therefrom. In other words, the DSP 30 is disposed between the application software side of the audio system that records/reproduces audio data and additional information and the memory card 40. The DSP 30 is operated when the memory card 40 is accessed. In addition, the DSP 30 is operated corresponding to software such as a file system.

[0023]

The DSP 30 manages files stored in the memory card 40 with the FAT system used in conventional personal computers. In addition to the file system, according to the embodiment of the present invention, a management file is used. The management file will be descried later. The management file is used to manage data files stored in the memory card 40. The management file as the first file management information is used to manage audio data files. On the other hand, the FAT as the second file management information is used to mange all files including audio

data files and management files stored in the flash memory of the memory card 40. The management file is stored in the memory card 40. The FAT is written to the flash memory along with the route directory and so forth before the memory card 40 is shipped. The details of the FAT will be described later.

[0024]

According to the embodiment of the present invention, to protect the copyright of data, audio data that has been compressed corresponding to the ATRAC3 format is encrypted. On the other hand, since it is not necessary to protect the copyright of the management file, it is not encrypted. There are two types of memory cards that are an encryption type and a non-encryption type. However, a memory card for use with the recorder that records copyright protected data is limited to the encryption type.

[0025]

Fig. 3 is a block diagram showing the internal structure of the memory card 40. The memory card 40 comprises a control block 41 and a flash memory 42 that are structured as a one-chip IC. A bidirectional serial interface is disposed between the DSP 30 of the recorder and the memory card 40. The bidirectional serial interface is composed of ten lines that are a clock line SCK for transmitting a clock signal that is transmitted along with data, a status

line SBS for transmitting a signal that represents a status, a data line DIO for transmitting data, an interrupt line INT, two GND lines, two INT lines, and two reserved lines.

[0026]

The clock line SCK is used for transmitting a clock signal in synchronization with data. The status line SBS is used for transmitting a signal that represents the status of the memory card 40. The data line DIO is used for inputting and outputting a command and encrypted audio data. The interrupt line INT is used for transmitting an interrupt signal that causes the memory card 40 to interrupt the DSP 30 of the recorder. When the memory card 40 is attached to the recorder, the memory card 40 generates the interrupt signal. However, according to the embodiment of the present invention, since the interrupt signal is transmitted through the data line DIO, the interrupt line INT is grounded.

[0027]

A serial/parallel converting, parallel/serial converting, and interface block (S/P, P/S, I/F block) 43 is an interface disposed between the DSP 30 of the recorder and the control block 41 of the memory card 40. The S/P, P/S, and IF block 43 converts serial data received from the DSP 30 of the recorder into parallel data and supplies the parallel data to the control

block 41. In addition, the S/P, P/S, and IF block 43 converts parallel data received from the control block 41 into serial data and supplies the serial data to the DSP 30. When the S/P, P/S, and IF block 43 receives a command and data through the data line DIO, the S/P, P/S, and IF block 43 separates them into these that are normally accessed to the flash memory 42 and those that are encrypted.

[0028]

In the format of which data is transmitted through the data line DIO, after a command is transmitted, data is transmitted. The S/P, P/S, and IF block 43 detects the code of a command and determines whether the command and data are those that are normally accessed or those that are encoded. Corresponding to the determined result, the S/P, P/S, and IF block 43 stores a command that is normally accessed to a command register 44 and stores data that is normally accessed to a page buffer 45 and a write register 46. In association with the write register 46, the memory card 40 has an error correction code encoding circuit 47. The error correction code encoding circuit 47 generates a redundant code that is an error correction code for data temporarily stored in the page buffer 45.

[0029]

Output data of the command register 44, the

page buffer 45, the write register 46, and the error correction code encoding circuit 47 is supplied to a flash memory interface and sequencer (hereinafter, referred to as memory I/F and sequencer) 51. The memory IF and sequencer 51 is an interface disposed between the control block 41 and the flash memory 42 and controls data exchanged therebetween. Data is written to the flash memory through the memory IF and sequencer 51.

[0030]

Audio data that has been compressed corresponding to the ATRAC3 format and written to the flash memory (hereinafter, this audio data is referred to as ATRAC3 data) is encrypted by the security IC 20 of the recorder/player and the security block 52 of the memory card 40 so as to protect the copyright of the ATRAC3 data. The security block 52 comprises a buffer memory 53, a DES encrypting circuit 54, and a nonvolatile memory 55.

[0031]

The security block 52 of the memory card 40 has a plurality of authentication keys and a unique storage key for each memory card. The nonvolatile memory 55 stores a key necessary for encrypting data. The key stored in the nonvolatile memory 55 cannot be analyzed. According to the embodiment, for example, a storage key is stored in the nonvolatile memory 55.

The security block 52 also has a random number generating circuit. The security block 52 authenticates an applicable recorder/player and shares a session key therewith. In addition, the security block 52 re-encrypts contents with the storage key through the DSE encrypting circuit 54.

[0032]

For example, when the memory card 40 is attached to the recorder, they are mutually authenticated. The security IC 20 of the recorder and the security block 52 of the memory card 40 mutually authenticate. When the recorder has authenticated the attached memory card 40 as an applicable memory card and the memory card 40 has authenticated the recorder as an applicable recorder, they are mutually authenticated. After the mutual authenticating process has been successfully performed, the recorder/player and the memory card 40 generate respective session keys and share them with each other. Whenever the recorder/player and the memory card 40 authenticate each other, they generate respective session keys.

[0033]

When contents are written to the memory card 40, the recorder/player encrypts a contents key with a session key and supplies the encrypted data to the memory card 40. The memory card 40 decrypts the contents key with the session key, re-encrypts the

contents key with a storage key, and supplies the contents key to the recorder. The storage key is a unique key for each memory card 40. When the recorder/player receives the encrypted contents key, the recorder/player performs a formatting process for the encrypted contents key, and writes the encrypted contents key and the encrypted contents to the memory card 40.

[0034]

Data that is read from the flash memory 42 is supplied to the page buffer 45, the read register 48, and the error correction circuit 49 through the memory IF and the sequencer 51. The error correcting circuit 49 corrects an error of the data stored in the page buffer 45. Output data of the page buffer 45 that has been error-corrected and the output data of the read register 48 are supplied to the S/P, P/S, and IF block 43. The output data of the S/P, P/S, and IF block 43 is supplied to the DSP 30 of the recorder through the above-described serial interface.

[0035]

When data is read from the memory card 40, the contents key encrypted with the storage key and the contents encrypted with the block key are read from the flash memory 42. The security block 52 decrypts the contents key with the storage key. The security block 52 re-encrypts the decrypted content key with the

session key and transmits the re-encrypted contents key to the recorder. The recorder decrypts the contents key with the received session key and generates a block key with the decrypted contents key. The recorder/player successively decrypts the encrypted ATRAC3 data.

[0036]

A configuration ROM 50 is a memory that stores partition information, various types of attribute information, and so forth of the memory card 40. The memory card 40 also has an erase protection switch 60. When the switch 60 is in the erase protection position, even if a command that causes the memory card 40 to erase data stored in the flash memory 42 is supplied from the recorder/player side to the memory card 40, the memory card 40 is prohibited from erasing the data stored in the flash memory 42. An OSC cont. 61 is an oscillator that generates a clock signal that is the reference of the timing of the process of the memory card 40.

[0037]

Fig. 4 is a schematic diagram showing the hierarchy of the processes of the file system of the computer system that uses a memory card as a storage medium. On the hierarchy, the top hierarchical level is an application process layer. The application process layer is followed by a file management process

layer, a logical address management layer, a physical address management layer, and a flash memory access layer. In the above-mentioned hierarchical structure, the file management process layer is the FAT file system. Physical addresses are assigned to individual blocks of the flash memory. The relation between the blocks of the flash memory and the physical addresses thereof does not vary. Logical addresses are addresses that are logically handled on the file management process layer.

[0038]

Fig. 5 is a schematic diagram showing the physical structure of data handled in the flash memory 42 of the memory card 40. In the memory 42, a data unit (referred to as segment) is divided into a predetermined number of blocks (fixed length). One block is divided into a predetermined number of pages (fixed length). In the flash memory, data is erased as each block at a time. Data is written to the flash memory 42 or read therefrom as a page at a time. The size of each block is the same. Likewise, the size of each page is the same. One block is composed of page 0 to page m. For example, one block has a storage capacity of for example 8 KB (kilobytes) or 16 KB. One page has a storage capacity of 512 B (bytes). When one block has a storage capacity of 8 KB, the total storage capacity of the flash memory 42 is 4 MB (512 blocks) or

8 MB (1024 blocks). When one block has a storage capacity of 16 KB, the total storage capacity of the flash memory 42 is 16 MB (1024 blocks), 32 MB (2048 blocks), or 64 MB (4096 blocks).

[0039]

One page is composed of a data portion of 512 bytes and a redundant portion of 16 bytes. The first three bytes of the redundant portion is an overwrite portion that is rewritten whenever data is updated. The first three bytes successively contain a block status area, a page status area, and an update status area. The remaining 13 bytes of the redundant portion are fixed data that depends on the contents of the data portion. The 13 bytes contain a management flag area (1 byte), a logical address area (2 bytes), a format reserve area (5 bytes), a dispersion information ECC area (2 bytes), and a data ECC area (3 bytes). The dispersion information ECC area contains redundant data for an error correction process against the management flag area, the logical address area, and the format reserve area. The data ECC area contains redundant data for an error correction process against 512-byte data.

[0040]

The management flag area contains a system flag (1: user block, 0: boot block), a conversion table flag (1: invalid, 0: table block), a copy prohibition

flag (1: OK, 0: NG), and an access permission flag (1: free, 0: read protect).

[0041]

The first two blocks - blocks 0 and 1 are boot blocks. The block 1 is a backup of the block 0. The boot blocks are top blocks that are valid in the memory card. When the memory card is attached to the recorder/player, the boot blocks are accessed at first. The remaining blocks are user blocks. Page 0 of the boot block contains a header area, a system entry area, and a boot and attribute information area. Page 1 of the boot block contains a prohibited block data area. Page 2 of the boot block contains a CIS (Card Information Structure)/IDI (identify Drive Information) area.

[0042]

The header area of the boot block contains a boot block ID and the number of effective entries. The system entries are the start position of prohibited block data, the data size thereof, the data type thereof, the data start position of the CIS/IDI area, the data size thereof, and the data type thereof. The boot and attribute information contains the memory card type (read only type, rewritable type, or hybrid type), the block size, the number of blocks, the number of total blocks, the security/non-security type, the card fabrication data (date of fabrication), and so forth.

[0043]

Since the flash memory has a restriction for the number of rewrite times due to the deterioration of the insulation film, it is necessary to prevent the same storage area (block) from being concentratedly accessed. Thus, when data at a particular logical address stored at a particular physical address is rewritten, updated data of a particular block is written to a non-used block rather than the original block. Thus, after data is updated, the relation between the logical address and the physical address changes. This process is referred to as swap process. Consequently, the same block is prevented from being concentratedly accessed. Thus, the service life of the flash memory can be prolonged.

[0044]

The logical address associates with data written to the block. Even if the block of the original data is different from the block of updated data, the address on the FAT does not change. Thus, the same data can be properly accessed. However, since the swap process is performed, a conversion table that correlates logical addresses and physical addresses is required (this table is referred to as logical-physical address conversion table). With reference to the logical-physical address conversion table, a physical address corresponding to a logical address designated

on the FAT is obtained. Thus, a block designated with a physical address can be accessed.

[0045]

The DSP 30 stores the logical-physical address conversion table in the SRAM. When the storage capacity of the RAM is small, the logical-physical address conversion table can be stored to the flash memory. The logical-physical address conversion table correlates logical addresses (2 bytes) sorted in the ascending order with physical addresses (2 bytes). Since the maximum storage capacity of the flash memory is 128 MB (8192 blocks), 8192 addresses can be assigned with two bytes. The logical-physical address conversion table is managed for each segment. Thus, the size of the logical-physical address conversion table is proportional to the storage capacity of the flash memory. When the storage capacity of the flash memory is 8 MB (two segments), two pages are used as the logical-physical address conversion table for each of the segments. When the conversion table is stored in the flash memory, a predetermined one bit of the management flag area in the redundant portion in each page represents whether or not the current block is a block containing the logical-physical address conversion table.

[0046]

The above-described memory card can be used

with the FAT file system of a personal computer system as with the disc shaped record medium. The flash memory has an IPL area, a FAT area, and a route directory area (not shown in Fig. 5). The IPL area contains the address of a program to be initially loaded to the memory of the recorder/player. In addition, the IPL area contains various types of memory information. The FAT area contains information with respect to blocks (clusters). The FAT has defined unused blocks, next block number, defective blocks, and last block number. The route directory area contains directory entries that are a file attribute, an update date [day, month, year], file size, and so forth.

[0047]

According to the embodiment of the present invention, in addition to the file management system defined in the format of the memory card 40, the management file is used for managing tracks and parts of music files. The management file is recorded to a user block of the flash memory 42 of the memory card 40. Thus, as will be described later, even if the FAT of the memory card 40 is destroyed, a file can be recovered.

[0048]

The management file is generated by the DSP 30. When the power of the recorder/player is turned on, the DSP 30 determines whether or not the memory

card 40 has been attached to the recorder/player. When the memory card has been attached, the DSP 30 authenticates the memory card 40. When the DSP 30 has successfully authenticated the memory card 40, the DSP 30 reads the boot block of the flash memory 42. Thus, the DSP 30 reads the physical-logical address conversion table and stores the read data to the SRAM. The FAT and the route directory have been written to the flash memory of the memory card 40 before the memory card 40 is shipped. When data is recorded to the memory card 40, the management file is generated.

[0049]

In other words, a record command issued by the remote controller of the user or the like is supplied to the DSP 30 from the external controller through the bus and the bus interface 32. The encoder/decoder IC 10 compresses the received audio data and supplies the resultant ATRAC3 data to the security IC 20. The security IC 20 encrypts the ATRAC3 data. The encrypted ATRAC3 data is recorded to the flash memory 42 of the memory card 40. Thereafter, the FAT and the management file are updated. Whenever a file is updated (in reality, whenever the recording process of audio data is completed), the FAT and the management file stored in the SRAMs 31 and 36 are rewritten. When the memory card 40 is detached or the power of the recorder/player is turned off, the FAT and

the management file that are finally supplied from the SRAMs 31 and 36 are recorded to the flash memory 42. Alternatively, whenever the recording process of audio data is completed, the FAT and the management file written in the flash memory 42 may be rewritten. When audio data is edited, the contents of the management file are updated.

[0050]

In the data structure according to the embodiment, additional information is contained in the management file. The additional information is updated and recorded to the flash memory 42. In another data structure of the management file, an additional information management file is generated besides the track management file. The additional information is supplied from the external controller to the DSP 30 through the bus and the bus interface 32. The additional information is recorded to the flash memory 42 of the memory card 40. Since the additional information is not supplied to the security IC 20, it is not encrypted. When the memory card 40 is detached from the recorder/player or the power thereof is turned off, the additional information is written from the SRAM of the DSP 30 to the flash memory 42.

[0051]

Fig. 6 is a schematic diagram showing the file structure of the memory card 40. As the file

structure, there are a still picture directory, a moving picture directory, a voice directory, a control directory, and a music (HIFI) directory. According to the embodiment, music programs are recorded and reproduced. Next, the music directory will be described. The music directory has two types of files. The first type is a reproduction management file BLIST.MSF (hereinafter, referred to as PBLIST). The other type is an ATRAC3 data file A3Dnnnn.MSA that stores encrypted music data. The music directory can stores up to 400 ATRAC3 data files. ATRAC3 data files are registered to the reproduction management file and generated by the recorder/player.

[0052]

Fig. 7 is a schematic diagram showing the structure of the reproduction management file. Fig. 8 is a schematic diagram showing the file structure of one ATRAC3 data file. The reproduction management file is a fixed-length file of 16 KB. An ATRAC3 data file is composed of an attribute header and an encrypted music data area for each music program. The attribute data has a fixed length of 16 KB. The structure of the attribute header is similar to that of the reproduction management file.

[0053]

The reproduction management file is composed of a header, a memory card name NM-1S (for one byte

code), a memory card name NM2-S (for two byte code), a program reproduction sequence table TRKTBL, and memory card additional information INF-S. The attribute header at the beginning of the data file is composed of a header, a program name NM1 (for one byte code), a program name NM2 (for two byte code), track information TRKINF (such as track key information), part information PRTINF, and track additional information INF. The header contains information of the number of total parts, the attribute of the name, the size of the additional information, and so forth.

[0054]

The attribute data is followed by ATRAC3 music data. The music data is block-segmented every 16 KB. Each block starts with a header. The header contains an initial value for decrypting encrypted data. Only music data of an ATRAC3 data file is encrypted. Thus, other data such as the reproduction management file, the header, and so forth are not encrypted.

[0055]

Next, with reference to Fig. 9, the relation between music programs and ATRAC3 data files will be described. One track is equivalent to one music program. In addition, one music program is composed of one ATRAC3 data (see Fig. 8). The ATRAC3 data file is audio data that has been compressed corresponding to

the ATRAC3 format. The ATRAC3 data file is recorded as a cluster at a time to the memory card 40. One cluster has a capacity of 16 KB. A plurality of files are not contained in one cluster. The minimum data erase unit of the flash memory 42 is one block. In the case of the memory card 40 for music data, a block is a synonym of a cluster. In addition, one cluster is equivalent to one sector.

[0056]

One music program is basically composed of one part. However, when a music program is edited, one music program may be composed of a plurality of parts. A part is a unit of data that is successively recorded. Normally, one track is composed of one part. The connection of parts of a music program is managed with part information PRTINF in the attribute header of each music program. In other words, the part size is represented with part size PRTSIZE (4 bytes) of the part information PRTINF. The first two bytes of the part size PRTSIZE represents the number of total clusters of the current part. The next two bytes represent the positions of the start sound unit (SU) and the end sound unit (SU) of the beginning and last clusters, respectively. Hereinafter, a sound unit is abbreviated as SU. With such a part notation, when music data is edited, the movement of the music data can be suppressed. When music data is edited for each

block, although the movement thereof can be suppressed, the edit unit of a block is much larger than the edit unit of a SU.

[0057]

SU is the minimum unit of a part. In addition, SU is the minimum data unit in the case that audio data is compressed corresponding to the ATRAC3 format. 1 SU is audio data of which data of 1024 samples at 44.1 kHz (1024 x 16 bits x 2 channels) is compressed to data that is around 10 times smaller than that of original data. The duration of 1 SU is around 23 msec. Normally, one part is composed of several thousand SU. When one cluster is composed of 42 SU, one cluster allows a sound of one second to be generated. The number of parts composing one track depends on the size of the additional information. Since the number of parts is obtained by subtracting the header, the program name, the additional data, and so forth from one block, when there is no additional information, the maximum number of parts (645 parts) can be used.

[0058]

Fig. 9 is a schematic diagram showing the file structure in the case that two music programs of a CD or the like are successively recorded. The first program (file 1) is composed of for example five clusters. Since one cluster cannot contain two files

of the first program and the second program, the file 2 starts from the beginning of the next cluster. Thus, the end of the part 1 corresponding to the file 1 is in the middle of one cluster and the remaining area of the cluster contains no data. Likewise, the second music program (file 2) is composed of one part. In the case of the file 1, the part size is 5. The first cluster starts at 0-th SU. The last cluster ends at 4-th SU.

[0059]

There are four types of edit processes that are a divide process, a combine process, an erase process, and a move process. The divide process is performed to divide one track into two portions. When the divide process is performed, the number of total tracks increases by one. In the divide process, one file is divided into two files on the file system. Thus, in this case, the reproduction management file and the FAT are updated. The combine process is performed to combine two tracks into one track. When the combine process is performed, the number of total tracks decreases by one. In the combine process, two files are combined into one file on the file system. Thus, when the combine process is performed, the reproduction management file and the FAT are updated. The erase process is performed to erase a track. The track numbers after the track that has been erased decrease one by one. The move process is performed to

change the track sequence. Thus, the reproduction management file and the FAT are updated. The other operation of the move process is to translate a track from memory card to other media, such as a hard disk. Comparing the copy process in which a replica of a track is generated, the move process means only translation of track position. Thus, a replica is not generated in performing the move process.

[0060]

Fig. 10 is a schematic diagram showing the combined result of two programs (file 1 and file 2) shown in Fig. 9. As a result of the combine process, the combined file is composed of two parts. Fig. 11 is a schematic diagram showing the divided result of which one program (file 1) is divided in the middle of the cluster 2. By the divide process, the file 1 is composed of clusters 0, 1, and the beginning portion of cluster 2. The file 2 is composed of the end portion of cluster 2 and clusters 3 and 4.

[0061]

As described above, according to the embodiment of the present invention, since the part notation is defined, as the combined result (see Fig. 10), the start position of the part 1, the end position of the part 1, and the end portion of the part 2 can be defined with SU. Thus, to pack the space due to the combined result, it is not necessary to move the music

data of the part 2. In addition, as the divided result (see Fig. 11), it is not necessary to move data and pack the space at the beginning of the file 2.

[0062]

Fig. 12 is a schematic diagram showing the detailed data structure of the reproduction management file PBLIST. Figs. 13A and 13B show a header portion and the remaining portion of the reproduction management file PBLIST. The size of the reproduction management file is one cluster (one block = 16 KB). The size of the header A is 32 bytes. The rest of the reproduction management file PBLIST contains a name NM1-S area (256 bytes) (for the memory card), a name NM2-S area (512 bytes), a contents key area, a MAC area, an S-YMDhms area, a reproduction sequence management table TRKTBL area (800 bytes), a memory card additional information INF-S area (14720 bytes), and a header information redundant area. The start positions of these areas are defined in the reproduction management file.

[0063]

The first 32 bytes of (0x0000) to (0x0010) are used for the header(see Fig. 13A). In the file, 16-byte areas are referred to as slots. The header are placed in the first and second slots. The header contains the following areas. An area denoted by "Reserved" is an undefined area. Normally, in a

reserved area, a null (0x00) is written. However, even if any data is written to a reserved area, the data written in the reserved is ignored. In a future version, some reserved areas may be used. In addition, data is prohibited from being written to a reserved area. When an option area is not used, it is treated as a reserved area.

[0064]

BLKID-TL0 (4 bytes)

Meaning: BLOCKID FILE ID

Function: Identifies the top of the reproduction management file.

Value: Fixed value = "TL = 0" (for example, 0x544C2D30)

MCode (2 bytes)

Meaning: MAKER CODE

Function: Identifies the maker and model of the recorder/player

Value: High-order 10 bits (Maker code); low-order 6 bits (model code).

REVISION (4 bytes)

Meaning: Number of rewrite times of PBLIST

Function: Increments whenever the reproduction management file is rewritten.

Value: Starts at 0 and increments by 1.

S-YMDhms (4 bytes) (Option)

Meaning: Year, month, day, hour, minute, and

second recorded by the recorder/player with a reliable clock.

Function: Identifies the last recorded date and time.

Value: bits 25 to 31: Year 0 to 99 (1980 to 2079)

bits 21 to 24: Month 0 to 12

bits 16 to 20: Day 0 to 31

bits 11 to 15: Hour 0 to 23

bits 05 to 10: Minute 0 to 59

bits 00 to 04: Second 0 to 29 (two bit interval)

[0065]

SY1C+L (2 bytes)

Meaning: Attribute of name (one byte code) of memory card written in NM1-S area.

Function: Represents the character code and the language code as one byte code.

Value: Character code (C): High-order one byte

00: Non-character code, binary number

01: ASCII (American Standard Code for Information Interchange)

02: ASCII+KANA

03: Modified 8859-1

81: MS-JIS

82: KS C 5601-1989

83: GB (Great Britain) 2312-80

90: S-JIS (Japanese Industrial Standards) (for Voice)

[0066]

Language code (L): Low-order one byte

Identifies the language based on EBU Tech 3258 standard.

00: Not set

08: German

09: English

0A: Spanish

0F: French

15: Italian

1D: Dutch

65: Korean

69: Japanese

75: Chinese

When data is not recorded, this area is all 0.

[0067]

SN2C+L (2 bytes)

Meaning: Attribute of name of memory card in NM2-S area.

Function: Represents the character code and the language coded as one byte code.

Value: Same as SN1C+L

SINFSIZE (2 bytes)

Meaning: Total size of additional information of memory card in INF-S area.

Function: Represents the data size as an increment of 16 bytes. When data is not recorded, this area is all 0.

Value: Size: 0x0001 to 0x39C (924)

T-TRK (2 bytes)

Meaning: TOTAL TRACK NUMBER

Function: Represents the number of total tracks.

Value: 1 to 0x0190 (Max. 400 tracks)

When data is recorded, this area is all 0.

VerNo (2 bytes)

Meaning: Format version number

Function: Represents the major version number (high order one byte) and the minor version number (low order one byte).

Value: 0x0100 (Ver 1.0)

0x0203 (Ver 2.3)

[0068]

Next, areas (see Fig. 13B) that preceded by the header will be described.

[0069]

NM1-S

Meaning: Name of memory card (as one byte code)

Function: Represents the name of the memory

card as one byte code (max. 256). At the end of this area, an end code (0x00) is written. The size is calculated from the end code. When data is not recorded, null (0x00) is recorded from the beginning (0x0020) of this area for at least one byte.

Value: Various character code

NM2-S

Meaning: Name of memory card (as two byte code)

Function: Represents the name of the memory card as two byte code (max. 512). At the end of this area, an end code (0x00) is written. The size is calculated from the end code. When data is not recorded, null (0x00) is recorded from the beginning (0x0120) of this area for at least two bytes.

Value: Various character code

[0070]

CONTENTS KEY

Meaning: Value for music program. Protected with MG(M) and stored. Same as CONTENTS KEY.

Function: Used as a key necessary for calculating MAC of S-YMDhms.

Value: 0 to 0xFFFFFFFFFFFFFFFF

MAC

Meaning: Forged copyright information check

value

Function: Represents the value generated with S-YMDhms and CONTENTS KEY.

Value: 0 to 0xFFFFFFFFFFFFFFFF

[0071]

TRK-nnn

Meaning: SQN (sequence) number of ATRAC3 data file reproduced.

Function: Represents FNo of TRKINF.

Value: 1 to 400 (0x190)

When there is no track, this area is all 0.

INF-S

Meaning: Additional information of memory card (for example, information with respect to photos, songs, guides, etc.)

Function: Represents variable length additional information with a header. A plurality of types of additional information may be used. Each of the types of additional information has an ID and a data size. Each additional information area including a header is composed of at least 16 bytes and a multiple of 4 bytes. For details, see the following section.

Value: Refer to the section of "Data Structure of Additional Information".

S-YMDhms (4 bytes) (Option)

Meaning: Year, month, day, hour, minute, and second recorded by the recorder/player with a reliable clock.

Function: Identifies the last recorded date and time. In this case of EMD, this area is mandatory.

Value: bits 25 to 31: Year 0 to 99 (1980 to 2079)

bits 21 to 24: Month 0 to 12

bits 16 to 24: Day 0 to 31

bits 11 to 15: Hour 0 to 23

bits 05 to 10: Minute 0 to 59

bits 00 to 04: Second 0 to 29 (two second interval)

[0072]

As the last slot of the reproduction management file, the same BLKID-TL0, MCode, and REVISION as those in the header are written.

[0073]

While data is being recorded to a memory card, it may be mistakenly or accidentally detached or the power of the recorder/player may be turned off. When such an improper operation is performed, a defect should be detected. As described above, the REVISION area is placed at the beginning and end of each block. Whenever data is rewritten, the value of the REVISION area is incremented. If a defect termination takes

place in the middle of a block, the value of the REVISION area at the beginning of the block does not match the value of the REVISION area at the end of the block. Thus, such a defect termination can be detected. Since there are two REVISION areas, the abnormal termination can be detected with a high probability. When an abnormal termination is detected, an alarm such as an error message is generated.

[0074]

In addition, since the fixed value BLKID-TL0 is written at the beginning of one block (16 KB), when the FAT is destroyed, the fixed value is used as a reference for recovering data. In other words, with reference to the fixed value, the type of the file can be determined. Since the fixed value BLKID-TL0 is redundantly written at the header and the end portion of each block, the reliability can be secured. Alternatively, the same reproduction management file can be redundantly recorded.

[0075]

The data amount of an ATRAC3 data file is much larger than that of the track information management file. In addition, as will be described later, a block number BLOCK SERIAL is added to ATRAC3 data file. However, since a plurality of ATRAC3 files are recorded to the memory card, to prevent them from become redundant, both CONNUM0 and BLOCK SERIAL are

used. Otherwise, when the FAT is destroyed, it will be difficult to recover the file.

[0076]

Likewise, the maker code (Mcode) is redundantly recorded at the beginning and the end of each block so as to identify the maker and the model in such a case that a file has been improperly recorded in the state that the FAT has not been destroyed.

[0077]

Fig. 13C is a schematic diagram showing the structure of the additional information data. The additional information is composed of the following header and variable length data. The header has the following areas.

[0078]

INF

Meaning: FIELD ID

Function: Represents the beginning of the additional information (fixed value).

Value: 0x69

ID

Meaning: Additional information key code

Function: Represents the category of the additional information.

Value: 0 to 0xFF

SIZE

Meaning: Size of individual additional

information

Function: Represents the size of each type of additional information. Although the data size is not limited, it should be at least 16 bytes and a multiple of 4 bytes. The rest of the data should be filled with null (0x00).

Value: 16 to 14784 (0x39C0)

MCode

Meaning: MAKER CODE

Function: Identifies the maker and model of the recorder/player.

Value: High-order 10 bits (maker code), low-order 10 bits (machine code).

C+L

Meaning: Attribute of characters in data area starting from byte 12.

Function: Represents the character code and the language code as one byte code.

Value: Same as SNC+L

DATA

Meaning: Individual additional information

Function: Represents each type of additional information with variable length data. Real data always starts from byte 12. The length (size) of the real data should be at least 4 bytes and a multiple of 4 bytes. The rest of the data area should be filled with null

(0x00).

Value: Individually defined corresponding to the contents of each type of additional information.

[0079]

Fig. 14 is a table that correlates key code values (0 to 63 of additional information and types thereof. Key code values (0 to 31) are assigned to music character information. Key code values (32 to 63) are assigned to URLs (Uniform Resource Locator) (web information). The music character information and URL information contain character information of the album title, the artist name, the CM, and so forth as additional information.

[0080]

Fig. 15 is a table that correlates key code values (64 to 127) of additional information and types thereof. Key code values (64 to 95) are assigned to paths/others. Key code values (96 to 127) are assigned to control/numeric data. For example, ID = 98 represents TOC-ID as additional information. TOC-ID represents the first music program number, the last music program number, the current program number, the total performance duration, and the current music program duration corresponding to the TOC information of a CD (Compact Disc).

[0081]

Fig. 16 is a table that correlates key code values (128 to 159) of additional information and types thereof. Key code values (128 to 159) are assigned to synchronous reproduction information. In Fig. 16, EMD stands for electronic music distribution.

[0082]

Next, with reference to Fig. 17, real examples of additional information will be described. As with Fig. 13C, Fig. 17A shows the data structure of the additional information. In Fig. 17B, key code ID = 3 (artist name as additional information). SIZE = 0x1C (28 bytes) representing that the data length of additional information including the header is 28 bytes; C+L representing that character code C = 0x01 (ASCII) and language code L = 0x09 (English). Variable length data after byte 12 represents one byte data "SIMON & GRAFUNKEL" as artist name. Since the data length of the additional information should be a multiple of 4 bytes, the rest is filled with (0x00).

[0083]

In Fig. 17C, key code ID = 97 representing that ISRC (International Standard Recording Code: Copyright code) as additional information. SIZE = 0x14 (20 bytes) representing that the data length of the additional information is 20 bytes. C = 0x00 and L = 0x00 representing that characters and language have not been set. Thus, the data is binary code. The variable

length data is eight-byte ISRC code representing copyright information (nation, copyright owner, recorded year, and serial number).

[0084]

In Fig. 17D, key code ID = 97 representing recorded date and time as additional information. SIZE = 0 x 10 (16 bytes) representing that the data length of the additional information is 16 bytes. C = 0x00 and L = representing that characters and language have not been set. The variable length data is four-byte code (32 bit) representing the recorded date and time (year, month, day, hour, minute, second).

[0085]

In Fig. 17E, key code ID = 107 representing a reproduction log as additional information. SIZE = 0x10 (16 bytes) representing that the data length of the additional information is 16 bytes. C = 0x00 and L = 0x00 representing that characters and language have not been set. The variable length data is a four-byte code representing a reproduction log (year, month, day, hour, minute, second). When the recorder/player has a reproduction log function, it records data of 16 bytes whenever it reproduces music data.

[0086]

Fig. 18 is a schematic diagram showing a data arrangement of ATRAC3 data file A3Dnnnn in the case that 1 SU is N bytes (for example, N = 384 bytes).

Fig. 18 shows an attribute header (1 block) of a data file and a music data file (1 block). Fig. 18 shows the first byte (0x0000 to 0x7FF0) of each slot of the two blocks (16 x 2 = 32k bytes). As shown in Fig. 18, the first 32 bytes of the attribute header are used as a header; 256 bytes are used as a music program area NM1 (256 bytes); and 512 bytes are used as a music program title area NM2 (512 bytes). The header of the attribute header contains the following areas.

[0087]

BLKID-HD0 (4 bytes)

Meaning: BLOCKID FIELD ID

Function: Identifies the top of an ATRA3 data file.

Value: Fixed value = "HD = 0" (For example, 0x48442D30)

MCode (2 bytes)

Meaning: MAKER CODE

Function: Identifies the maker and model of the recorder/player

Value: High-order 10 bits (maker code); low-order 6 bits (machine code)

BLOCK SERIAL (4 bytes)

Meaning: Track serial number

Function: Starets from 0 and increments by 1. Even if a music program is edited, this value does not vary.

Value: 0 to 0xFFFFFFFF.

[0088]

N1C+L (2 bytes)

Meaning: Represents the attribute of data (NM1) of a track (music program title).

Function: Represent the character code and language code of NM1 as one byte code.

Value: Same as SN1C+L

N2C+L (2 bytes)

Meaning: Represents the attribute of data (NM2) of a track (music program title).

Function: Represent the character code and language code of NM1 as one byte code.

Value: Same as SN1C+L

INFSIZE (2 bytes)

Meaning: Total size of additional information of current track.

Function: Represents the data size as a multiple of 16 bytes. When data is not recorded, this area should be all 0.

Value: 0x0000 to 0x3C6 (966)

T-PRT (2 bytes)

Meaning: Number of total bytes

Function: Represents the number of parts that composes the current track. Normally, the value of T-PRT is 1.

Value: 1 to 285 (645 dec).

T-SU (4 bytes)

Meaning: Number of total SU.

Function: Represents the total number of SU in one track that is equivalent to the program performance duration.

Value: 0x01 to 0x001FFFFF

INX (2 bytes) (Option)

Meaning: Relative position of INDEX

Function: Used as a pointer that represents the top of a representative portion of a music program. The value of INX is designated with a value of which the number of SU is divided by 4 as the current position of the program. This value of INX is equivalent to 4 times larger than the number of SU (around 93 msec).

Value: 0 to 0xFFFF (max, around 6084 sec)

XT (2 bytes) (Option)

Meaning: Reproduction duration of INDEX

Function: Designates the reproduction duration designated by INX-nnn with a value of which the number of SU is divided by 4. The value of INDEX is equivalent to four times larger than the normal SU (around 93 msec).

Value: 0x0000 (no setting); 0x01 to 0xFFFE (up to 6084 sec); 0xFFFF (up to end of music

program)

[0089]

Next, the music program title areas NM1 and NM2 will be described.

[0090]

NM1

Means: Character string of music program title

Function: Represents a music program title as one byte code (up to 256 characters) (variable length). The title area should be completed with an end code (0x00). The size should be calculated from the end code. When data is not recorded, null (0x00) should be recorded from the beginning (0x0020) of the area for at least one byte.

Value: Various character codes

NM2

Means: Character string of music program title

Function: Represents a music program title as two byte code (up to 512 characters) (variable length). The title area should be completed with an end code (0x00). The size should be calculated from the end code. When data is not recorded, null (0x100) should be recorded from the beginning (0x0120) of the

area for at least two bytes.

Value: Various character codes

[0091]

Data of 80 bytes starting from the fixed position (0x320) of the attribute header is referred to as track information area TRKINF. This area is mainly used to totally manage the security information and copy control information. Fig. 20 shows a part of TRKINF. The area TRKINF contains the following areas.

[0092]

CONTENTS KEY (8 bytes)

Meaning: Value for each music program. The value of CONTENTS KEY is protected in the security block of the memory card and then stored.

Function: Used as a key for reproducing a music program. It is used to calculate the value of MAC.

Value: 0 to 0xFFFFFFFFFFFFFFFF

MAC (8 bytes)

Meaning: Forged copyright information check value

Function: Represents the value generated with a plurality of values of TRKINF including contents cumulation numbers and a secret sequence number.

The secret sequence number is a sequence

number recorded in the secret area of the memory card. A non-copyright protection type recorder cannot read data from the secret area of the memory card. On the other hand, a copyright protection type recorder and a computer that operates with a program that can read data from a memory card can access the secret area.

[0093]

A (1 byte)

Meaning: Attribute of part.

Function: Represents the information of such as compression mode of a part.

Value: The details will be described in the following (see Fig. 21).

Next, the value of the area A will be described. In the following description, monaural mode (N = 0 or 1) is defined as a special joint mode of which bit 7 = 1, sub signal = 0, main signal = (L+R). A non-copyright protection type player may ignore information of bits 2 and 1.

[0094]

Bit 0 of the area A represents information of emphasis on/off state. Bit 1 of the area A represents information of reproduction skip or normal reproduction. Bit 2 of the area A represents information of data type such as audio data, FAX data, or the like. Bit 3 of the area A is undefined. By a combination of bits 4, 5, and 6, mode information of

ATRAC3 is defined as shown in Fig. 21. In other words, N is a mode value of 3 bits. For five types of modes that are monaural (N = 0 or 1), LP (N = 2), SP (N = 4), EX (N = 5), and HQ (N = 7), record duration (64 MB memory card only), data transmission rate, and the number of SU per block are listed. The number of bytes of 1 SU depends on each mode. The number of bytes of 1 SU in the monaural mode is 136 bytes. The number of bytes of 1 SU in the LP mode is 192 bytes. The number of bytes of 1 SU in the SP mode is 304 bytes. The number of bytes of 1 SU in the EX mode is 384 bytes. The number of bytes of 1 SU in the HQ mode is 512 bytes. Bit 7 of the area A represents ATRAC3 modes (0: Dual, 1: Joint).

[0095]

For example, an example of which a 64 MB memory card is used in the SP mode will be described. A 64-MB memory card has 3968 blocks. In the SP mode, since 1 SU is 304 bytes, one block has 53 SU. 1 SU is equivalent to (1024/44100) seconds. Thus, one block is (1024/44100) x 53 x (3968 - 10) = 4863 seconds = 81 minutes. The transmission rate is (44100/1024) x 304 x 8 = 104737 bps.

[0096]

LT (1 byte)

Meaning: Reproduction restriction flag (bits 7 and 6) and security partition (bits 5 to

0).

Function: Represents a restriction of the current track.

Value: bit 7: 0 = no restriction, 1 = restriction

bit 6: 0 = not expired, 1 = expired

bits 5 to 0: security partition (reproduction prohibited other than 0)

FNo (2 bytes)

Meaning: File number.

Function: Represents the initially recorded track number that designates the position of the MAC calculation value recorded in the secret area of the memory card.

Value: 1 to 0x190 (400)

MG(D) SERIAL-nnn (16 bytes)

Meaning: Represents the serial number of the security block (security IC 20) of the recorder.

Function: Unique value for each recorder

Value: 0 to 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

CONNUM (4 bytes)

Meaning: Contents cumulation number

Function: Represents a unique value cumulated for each music program. The value is managed by the security block of the

recorder/player. The upper limit of the value is 2^{32} that is 4,200,000,000. Used to identify a recorded program.

[0097]

Value: 0 to 0xFFFFFFFF

[0098]

YMDhms-S (4 bytes) (Option)

Meaning: Reproduction start date and time of track with reproduction restriction

Function: Represents the date and time at which data reproduction is permitted with EMD.

Value: Same as the notation of date and time of other areas

YMDhms-E (4 bytes) (Option)

Meaning: Reproduction end date and time of track with reproduction restriction

Function: Represents the date and time at which data reproduction is expired with EMD.

Value: Same as the notation of date and time of other areas

MT (1 byte) (Option)

Meaning: Maximum value of number of permitted reproduction times

Function: Represents the maximum number of reproduction times designated by EMD.

Value: 1 to 0xFF. When not used, the value

of the area MT is 00.

CT (1 byte) (Option)

Meaning: Number of reproduction times

Function: Represents the number of reproduction times in the number of permitted reproduction times. Whenever data is reproduced, the value of the area CT is gradually reduced.

Value: 0x00 to 0xFF. When not used, the value of the area CT is 0x00. When bit 7 of the area LT is 1 and the value of the area CT is 00, data is prohibited from being reproduced.

[0099]

CC (1 byte)

Meaning: COPY CONTROL

Function: Controls the copy operation.

Value: bits 6 and 7 represent copy control information. bits 4 and 5 represent copy control information of a high speed digital copy operation. bits 2 and 3 represent a security block authentication level. bits 0 and 1 are undefined.

Example of CC:

(bits 7 and 6)

11: Unlimited copy operation permitted

01: copy prohibited

00: one time copy operation permitted

(bits 3 and 2)

00: analog/digital input recording

MG authentication level is 0.

When digital record operation using data from a CD is performed, (bits 7 and 6): 00 and (bits 3 and 2): 00.

CN (1 byte) (Option)

Meaning: Number of permitted copy times in high speed serial copy management system

Function: Extends the copy permission with the number of copy times, not limited to one time copy permission and copy free permission. Valid only in first copy generation. The value of the area CN is gradually reduced whenever the copy operation is performed.

Value:

00: Copy prohibited

01 to 0xFE: Number of times

0xFF: Unlimited copy times

[0100]

The track information area TRKINF is followed by a 24-byte part management information area (PRTINF) starting from 0x0370. When one track is composed of a plurality of parts, the values of areas PRTINF of the individual parts are successively arranged on the time

axis. Fig. 23 shows a part of the area PRTINF. Next, areas in the area PRTINF will be described in the order of the arrangement.

[0101]

PRTSIZE (4 bytes)

Meaning: Part size

Function: Represents the size of a part. Cluster: 2 bytes (highest position), start SU: 1 byte (upper), end SU: 1 byte (lowest position).

Value: cluster: 1 to 0x1F40 (8000)

start SU: 0 to 0xA0 (160)

end SU: 0 to 0xA0 (16) (Note that SU starts from 0.)

PRTKEY (8 bytes)

Meaning: Part encrypting value

Function: Encrypts a part. Initial value = 0. Note that edit rules should be applied.

Value: 0 to 0xFFFFFFFFFFFFFFFF

CONNUM0 (4 bytes)

Meaning: Initially generated contents cumulation number key.

Function: Uniquely designates an ID of contents.

Value: Same value as the value of the contents cumulation number initial value key.

[0102]

As shown in Fig. 18, the attribute header of an ATRAC3 data file contains additional information INF. The additional information is the same as the additional information INF-S (see Figs. 12 and 13B) of the reproduction management file except that the start position is not fixed. The last byte position (a multiple of four bytes) at the end of one or a plurality of parts is followed by data of the additional information INF.

[0103]

INF

Meaning: Additional information with respect to track

Function: Represents variable length additional information with a header. A plurality of different types of additional information may be arranged. Each of additional information areas has an ID and a data size. Each additional information area is composed of at least 16 bytes and a multiple of 4 bytes.

Value: Same as additional information INF-S of reproduction management file

[0104]

The above-described attribute header is followed by data of each block of an ATRAC3 data file. As shown in Fig. 24, a header is added for each block.

Next, data of each block will be described.

[0105]

BLKID-A3D (4 bytes)

Meaning: BLOCKID FILE ID

Function: Identifies the top of ATRAC3 data.

Value: Fixed value = "A3D" (for example, 0x41334420)

Mcode (2 bytes)

Meaning: MAKER CODE

Function: Identifies the maker and model of the recorder/player

Value: High-order 10 bits (maker code); low-order 6 bits (model code)

CONNUMO (4 bytes)

Meaning: Cumulated number of initially created contents

Function: Designates a unique ID for contents. Even if the contents are edited, the value of the area CONNUMO is not changed.

Value: Same as the contents cumulation number initial key

BLOCK SERIAL (4 bytes)

Meaning: Serial number assigned to each track

Function: Starts from 0 and increments by 1. Even if the contents are edited, the value of the area BLOCK SERIAL is not changed.

Value: 0 to 0xFFFFFFFF

BLOCK-SEED (8 bytes)

Meaning: Key for encrypting one block

Function: The beginning of the block is a random number generated by the security block of the recorder/player. The random number is followed by a value incremented by 1. When the value of the area BLOCK-SEED is lost, since sound is not generated for around one second equivalent to one block, the same data is written to the header and the end of the block. Even if the contents are edited, the value of the area BLOCK-SEED is not changed.

Value: Initially 8-bit random number

INITIALIZATION VECTOR (8 bytes)

Meaning: Value necessary for encrypting/decrypting ATRAC3 data

Function: Represents an initial value necessary for encrypting and decrypting ATRAC3 data for each block. A block starts from 0. The next block starts from the last encrypted 8-bit value at the last SU. When a block is divided, the last eight bytes just before the start SU is used. Even if the contents are edited, the value of the area INITIALIZATION VECTOR is not changed.

Value: 0 to 0xFFFFFFFFFFFFFFFF

SU-nnn

Meaning: Data of sound unit

Function: Represents data compressed from 1024 samples. The number of bytes of output data depends on the compression mode. Even if the contents are edited, the value of the area SU-nnn is not changed. For example, in the SP mode, N = 384 bytes.

Value: Data value of ATRAC3

[0106]

In Fig. 18, since N = 384, 42 SU are written to one block. The first two slots (4 bytes) of one block are used as a header. In the last slot (two bytes), the areas BLKID-A3D, MCode, CONNUM0, and BLOCK SERIAL are redundantly written. Thus, M bytes of the remaining area of one block is (16,384 - 384 x 42 - 16 x 3 = 208) bytes. As described above, the eight-byte area BLOCK SEED is redundantly recorded.

[0107]

Next, a management file according to a second embodiment of the present invention will be described. Fig. 25 shows the file structure according to the second embodiment of the present invention. Referring to Fig. 25, a music directory contains a track information management file TRKLIST.MSF (hereinafter, referred to as TRKLIST), a backup track information management file TRKLISTB.MSF (hereinafter, referred to

as TRKLISTB), an additional information file INFLIST.MSF (that contains an artist name, an ISRC code, a time stamp, a still picture data, and so forth (this file is referred to as INFIST)), an ATRAC3 data file A3Dnnnn.MSF (hereinafter, referred to as A3nnnn). The file TRKLIST contains two areas NAME1 and NAME2. The area NAME1 is an area that contains the memory card name and the program name (for one byte code corresponding to ASCII/8859-1 character code). The area NAME2 is an area that contains the memory card name and the program name (for two byte code corresponding to MS-JIS/Hankul/Chinese code).

[0108]

Fig. 26 shows the relation between the track information management file TRKLIST, the areas NAME1 and NAME2, and the ATRAC3 data file A3Dnnnn. The file TRKLIST is a fixed-length file of 64k bytes (= 16 k x 4). An area of 32k bytes of the file is used for managing tracks. The remaining area of 32k bytes is used to contain the areas NAME1 and NAME2. Although the areas NAME1 and NAME2 for program names may be provided as a different file as the track information management file, in a system having a small storage capacity, it is convenient to totally manage the track information management file and program name files.

[0109]

The track information area TRKINF-nnnn and

part information area PRTINF-nnnn of the track information management file TRKLIST are used to manage the data file A3Dnnnn and the additional information INFLIST. Only the ATRAC3 data file A3Dnnnn is encrypted. In Fig. 26, the data length in the horizontal direction is 16 bytes (0 to F). A hexadecimal number in the vertical direction represents the value at the beginning of the current line.

[0110]

According to the second embodiment, three files that are the track management file TRKLIST (including a program title file), the additional information management file INFLIST, and the data file A3Dnnnn are used. According to the first embodiment (see Figs. 6, 7, and 8), two files that are the reproduction management file PBLIST for managing all the memory card and the data file ATRAC3 for storing programs are used.

[0111]

Next, the data structure according to the second embodiment will be described. For simplicity, in the data structure according to the second embodiment, the description of similar portions to those of the first embodiment is omitted.

[0112]

Fig. 27 shows the detailed structure of the track information management file TRKLIST. In the

track information management file TRKLIST, one cluster (block) is composed of 16k bytes. The size and data of the file TRKLISTB are the same as those of the backup file TRKLISTB. The first 32 bytes of the track information management file are used as a header. As with the header of the reproduction management file PBLIST, the header of the file TRKLIST contains a BLKID-TL0/TL1 (backup file ID) area (4 bytes), an area T-TRK (2 bytes) for the number of total tracks, a maker code area MCode (2 bytes), an area REVISION (4 bytes) for the number of TRKLIST rewrite times, and an area S-YMDhms (4 bytes) (option) for update date and time data. The meanings and functions of these data areas are the same as those of the first embodiment. In addition, the file TRKLIST contains the following areas.

[0113]

YMDhms (4 bytes)

Represents the last update date (year, month, day) of the file TRKLIST.

N1 (1 byte) (Option)

Represents the sequential number of the memory card (numerator side). When one memory card is used, the value of the area N1 is 0x01.

N2 (1 byte) (Option)

Represents the sequential number of the memory card (denominator side). When one memory card

is used, the value of the area N2 is 0x01.

MSID (2 bytes) (Option)

Represents the ID of a memory card. When a plurality of memory cards is used, the value of the area MSID of each memory card is the same (T.B.D.). (T.B.D. (to be defined) represents that this value may be defined in future).

S-TRK (2 bytes).

Represents a special track (T.B.D.). Normally, the value of the area S-TRK is 0x0000.

PASS (2 bytes) (Option)

Represents a password (T.B.D.).

APP (2 bytes) (Option)

Represents the definition of a reproduction application (T.B.D.) (normally, the value of the area APP is 0x0000).

INF-S (2 bytes) (Option)

Represents the additional information pointer of the entire memory card. When there is no additional information, the value of the area INF-S is 0x00.

[0114]

The last 16 bytes of the file TRKLIST are used for an area BLKID-TL0, an area MCode, and an area REVISION that are the same as those of the header. The backup file TRKLISTB contains the above-described header. In this case, the header contains an area BLKID-TL1, an area MCode, and an area REVISION.

[0115]

The header is followed by a track information area TRKINF for information with respect to each track and a part information area PRTINF for information with respect to each part of tracks (music programs). Fig. 27 shows the areas preceded by the area TRKLIST. The lower portion of the area TRKLISTB shows the detailed structure of these areas. In Fig. 27, a hatched area represents an unused area.

[0116]

The track information area TRKINF-nnn and the part information area PRTINF-nnn contain areas of an ATRAC3 data file. In other words, the track information area TRKINF-nnn and the part information area PRTINF-nnn each contain a reproduction restriction flag area LT (1 byte), a contents key area CONTENTS KEY (8 bytes), a recorder/player security block serial number area MG(D) SERIAL (16 bytes), an area XT (2 bytes) (option) for representing a feature portion of a music program, an area INX (2 bytes) (option), an area YMDhms-S (4 bytes) (option), an area YMDhms-E (4 bytes) (option), an area MT (1 byte) (option), an area CT (1 byte) (option), an area CC (1 byte) (option), an area CN (1 byte) (option) (these areas YMDhms-S, YMDhms-E, MT, CT, CC, and CN are used for reproduction restriction information and copy control information), an area A (1 byte) for part attribute, a part size area

PRTSIZE (4 bytes), a part key area PRTKEY (8 bytes), and a contents cumulation number area CONNUM (4 bytes). The meanings, functions, and values of these areas are the same as those of the first embodiment. In addition, the track information area TRKINF-nnn and the part information area PRTINF-nnn each contain the following areas.

[0117]

T0 (1 byte)

Fixed value (T0 = 0x74)

INF-nnn (Option) (2 bytes)

Represents the additional information pointer (0 to 409) of each track. 00: music program without additional information.

FNM-nnn (4 bytes)

Represents the file number (0x0000 to 0xFFFF) of an ATRK3 data file.

The number nnnn (in ASCII) of the ATRAC3 data file name (A3Dnnnn) is converted into 0xnnnnn.

APP_CTL (4 bytes) (Option)

Represents an application parameter (T.B.D.) (Normally, the value of the area APP_CTL is 0x0000).

P-nnn (2 bytes)

Represents the number of parts (1 to 2039) that compose a music program. This area corresponds to the above-described area T-PART.

PR (1 byte)

Fixed value (PR = 0 x 50).

[0118]

Next, the areas NAME1 (for one byte code) and NAME2 (for two byte code) for managing names will be described. Fig. 28 shows the detailed structure of the area NAME1 (for one byte code area). Each of the areas NAME1 and NAME2 (that will be described later) is segmented with eight bytes. Thus, their one slot is composed of eight bytes. At 0x8000 that is the beginning of each of these areas, a header is placed. The header is followed by a pointer and a name. The last slot of the area NAME1 contains the same areas as the header.

[0119]

BLKID-NM1 (4 bytes)

Represents the contents of a block (fixed value) (NM1 = 0x4E4D2D31).

PNM1-nnn (4 bytes) (Option)

Represents the pointer to the area NM1 (for one byte code).

PNM1-S

Represents the pointer to a name representing a memory card.

nnn (= 1 to 408) represents the pointer to a music program title.

The pointer represents the start position (2 bytes) of the block, the character code type (2 bits),

and the data size (14 bits).

NM1-nnn (Option)

Represents the memory card name and music program title for one byte code (variable length). An end code (0x00) is written at the end of the area.

[0120]

Fig. 29 shows the detailed data structure of the area NAME2 (for two byte code). At 0x8000 that is the beginning of the area, a header is placed. The header is followed by a pointer and a name. The last slot of the area NAME2 contains the same areas as the header.

[0121]

BLKID-NM2 (4 bytes)

Represents the contents of a block (fixed value) (NM2 = 0x4E4D2D32).

PNM2-nnn (4 bytes) (Option)

Represents the pointer to the area NM2 (for two byte code).

PNM2-S represents the pointer to the name representing the memory card. nnn (= 1 to 408) represents the pointer to a music program title.

The pointer represents the start position (2 bytes) of the block, the character code type (2 bits), and the data size (14 bits).

NM2-nnn (Option)

Represents the memory card name and music

program title for two byte code (variable). An end code (0x0000) is written at the end of the area.

[0122]

Fig. 30 shows the data arrangement (for one block) of the ATRAC3 data file A3Dnnnn in the case that 1 SU is composed of N bytes. In this file, one slot is composed of eight bytes. Fig. 30 shows the values of the top portion (0x0000 to 0x3FF8) of each slot. The first four slots of the file are used for a header. As with the data block preceded by the attribute header of the data file (see Fig. 18) of the first example, a header is placed. The header contains an area BLKID-A3D (4 bytes), a maker code area MCode (2 bytes), an area BLOCK SEED (8 bytes) necessary for encrypting process, an area CONNUM0 (4 bytes) for the initial contents cumulation number, a serial number area BLOCK SERIAL (4 bytes) for each track, and an area INITIALIZATION VECTOR (8 bytes) necessary for encrypting/decrypting process. The second last slot of the block redundantly contains an area BLOCK SEED. The last slot contains areas BLKID-A3D and MCode. As with the first embodiment, the header is followed by the sound unit data SU-nnnn.

[0123]

Fig. 31 shows the detailed data structure of the additional information management file INFLIST that contains additional information. In the second

embodiment, at the beginning (0x0000) of the file INFLIST, the following header is placed. The header is followed by the following pointer and areas.

[0124]

BLKID-INF (4 bytes)

Represents the contents of the block (fixed value) (INF = 0x494E464F).

T-DAT (2 blocks)

Represents the number of total data areas (0 to 409).

MCode (2 bytes)

Represents the maker code of the recorder/player

YMDhms (4 bytes)

Represents the record updated date and time.

INF-nnnn (4 bytes)

Represents the pointer to the area DATA of the additional information (variable length, as 2 bytes (slot) at a time). The start position is represented with the high order 16 bits (0000 to FFFF).

DataSlot-0000 (0x0800)

Represents the offset value from the beginning (as a slot at a time).

The data size is represented with low order 16 bits (0001 to 7FFF). A disable flag is set at the most significant bit. MSB = 0 (Enable), MSB = 1 (Disable)

The data size represents the total data amount of the music program.

(The data starts from the beginning of each slot. (The non-data area of the slot is filled with 00.)

The first INF represents a pointer to additional information of the entire album (normally, INF-409).

[0125]

Fig. 32 shows the structure of additional information. An 8-byte header is placed at the beginning of one additional information data area. The structure of the additional information is the same as that of the first embodiment (see Fig. 13C). In other words, the additional information contains an area IN (2 bytes) as an ID, an area key code ID (1 byte), an area SIZE (2 bytes) that represents the size of each additional information area, and a maker code area MCode (2 bytes). In addition, the additional information contains an area SID (1 byte) as a sub ID.

[0126]

According to the second embodiment of the present invention, in addition to the file system defined as a format of the memory card, the track information management file TRKLIST for music data is used. Thus, even if the FAT is destroyed, the file can be recovered. Fig. 33 shows a flow of a file

recovering process. To recover the file, a computer that operates with a file recovery program and that can access the memory card and a storing device (hard disk, RAM, or the like) connected to the computer are used. The computer has a function equivalent to the DSP30. Next, a file recovering process using the track management file TRKLIST will be described.

[0127]

All blocks of the flash memory whose FAT has been destroyed are searched for TL-0 as the value (BLKID) at the top position of each block. In addition, all the blocks are searched for NM-1 as the value (BLKID) at the top position of each block. Thereafter, all the blocks are searched for NM-2 as the value (BLKID) at the top position of each block. All the contents of the four blocks (track information management file) are stored to for example a hard disk by the recovery computer.

[0128]

The number of total tracks is obtained from data after the fourth byte of the track information management file. The 20-th byte of the track information area TRKINF-001, the value of the area CONNUM-001 of the first music program, and the value of the next area P-001 are obtained. The number of parts is obtained with the value of the area P-001. The values of the areas PRTSIZE of all parts of the track 1

of the area PRTINF is obtained. The number of total blocks (clusters) n is calculated and obtained.

[0129]

After the track information management file is obtained, the flow advances to step 102. At step 102, a voice data file (ATRAC3 data file) is searched. All blocks of other than the management file is searched from the flash memory. Blocks whose top value (BLKID) is A3D are collected.

[0130]

A block of which the value of the area CONNUM0 at the 16-th byte of A3Dnnnn is the same as that of the area CONNUM-001 of the first music program of the track information management file and of which the value of the area BLOCK SERIAL that starts from 20-th byte is 0 is searched. After the first block is obtained, a block (cluster) with the same value of the area CONNUM value as the first block and of which the value of BLOCK SERIAL is incremented by 1 (1 = 0 + 1) is searched. After the second block is obtained, a block with the same value of the area CONNUM0 as the second block and of which the value of the area BLOCK SERIAL is incremented by 1 (2 = 1 + 1) is searched.

[0131]

By repeating the process, the ATRC3 data file is searched until n blocks (clusters) of the track 1 are obtained. When all the blocks (clusters) are

obtained, they are successively stored to the hard disk.

[0132]

The same process for the track 1 is performed for the track 2. In other words, a block of which the value of the area CONNUM0 is the same as that of the area CONNUM-002 of the first music program of the track information management file and of which the value of the area BLOCK SERIAL that starts at the 20-th byte is searched. Thereafter, in the same manner as the track 1, the ATRAC3 data file is searched until the last block (cluster) n' is detected. After all blocks (clusters) are obtained, they are successively stored to the hard disk.

[0133]

By repeating the above-described process for all tracks (the number of tracks: m), all the ATRAC3 data is stored to the hard disk controlled by the recovering computer.

[0134]

At step 103, the memory card whose the FAT has been destroyed is re-initialized and then the FAT is reconstructed. A predetermined directory is formed in the memory card. Thereafter, the track information management file and the ATRAC3 data file for m tracks are copied from the hard disk to the memory card. Thus, the recovery process is finished.

[0135]

In the management file and data file, important parameters (in particular, codes in headers) may be recorded triply rather than doubly. When data is redundantly recorded, the same data may be recorded at any positions as long as they are apart from each other for one page or more.

[0136]

[Effects of the Invention]

According to the present invention, in addition to file management information defined in the nonvolatile memory such as the FAT, another (second) file management information is generated and stored in the nonvolatile memory. In the second file management information, a fixed length identification code that represents file management information is added. Thus, even if the FAT is destroyed, a file can be easily recovered with the file management information. According to the present invention, since the file management information has a fixed length identification value, the efficiency of the recovering process can be improved. Thus, it is not necessary for the user to make a backup file.

[0137]

Moreover, in the file management information, important parameters are redundantly recorded. Thus, important parameters can be securely protected. In

addition, since information that represents the number of rewrite times of a file is recorded at a position apart from the other by 1 page unit or more, a trouble in the middle of the rewriting process of a file can be detected. Moreover, when a trouble takes place, the cause of the trouble can be easily obtained.

[0138]

According to the present invention, in addition to the concept of the file, part management information is stored. Even if one track (music program) is composed of a plurality of parts, they can be easily managed. Moreover, since the part management information for parts that compose a track is handled along with the track management information (TRKINF) for tracks, the process can be more easily performed than that of Mini-Disc using links (Link-P).

[Brief Description of the Drawings]

[Fig. 1]

Block diagram showing the structure of the embodiment of the prevent invention.

[Fig. 2]

Block diagram showing the internal structure of a DSP in the embodiment of the prevent invention.

[Fig. 3]

Block diagram showing the internal structure of a memory card in the embodiment of the prevent invention.

[Fig. 4]

Schematic diagram showing a data structure of a file system operating layer in the flash memory in the embodiment of the prevent invention.

[Fig. 5]

Schematic diagram showing the physical structure of data in a flash memory in the embodiment of the present invention.

[Fig. 6]

Schematic diagram showing the regulation of file in the embodiment of the present invention.

[Fig. 7]

Schematic diagram showing the data structure of a reproduction management file in the embodiment of the present invention.

[Fig. 8]

Schematic diagram showing the data structure of a data file in the embodiment of the present invention.

[Fig. 9]

Schematic diagram showing the structure of data file in the embodiment of the present invention.

[Fig. 10]

Schematic diagram showing one example of edit operation to data file in the embodiment of the present invention.

[Fig. 11]

Schematic diagram showing another example of edit operation to data file in the embodiment of the present invention.

[Fig. 12]

Schematic diagram showing the structure of a reproduction management file in the embodiment of the present invention.

[Fig. 13]

Schematic diagram showing the structure of a reproduction management file and additional information area.

[Fig. 14]

Schematic diagram showing a example of additional data in the embodiment of the present invention.

[Fig. 15]

Schematic diagram showing a example of additional data in the embodiment of the present invention.

[Fig. 16]

Schematic diagram showing a example of additional data in the embodiment of the present invention.

[Fig. 17]

Schematic diagram showing concrete data structure of additional data in the embodiment of the present invention.

[Fig. 18]

Schematic diagram showing the data structure of data file in the embodiment of the present invention.

[Fig. 19]

Schematic diagram showing the one part of attribute header of data file.

[Fig. 20]

Schematic diagram showing the one part of attribute header of data file.

[Fig. 21]

Schematic diagram showing the kinds of record mode and record period in various record mode.

[Fig. 22]

Schematic diagram showing the copy control information

[Fig. 23]

Schematic diagram showing a part of the attribute header.

[Fig. 24]

Schematic diagram showing headers for each data blocks in the data file.

[Fig. 25]

Schematic diagram showing the regulation of another data structure that can be used in the present invention.

[Fig. 26]

Schematic diagram showing the relationship between files in another data structure.

[Fig. 27]

Schematic diagram showing the structure of the track management file.

[Fig. 28]

Schematic diagram showing the name file in the track management file in another data structure.

[Fig. 29]

Schematic diagram showing the name file in the track management file in another data structure.

[fig. 30]

Schematic diagram showing the data structure of data file in another data structure.

[fig. 31]

Schematic diagram showing the data structure of additional information management file.

[fig. 32]

Schematic diagram showing the data structure of additional information data.

[fig. 33]

Schematic diagram showing the process to recover the file.

[Description of Reference Numerals]

10 ... Audio encoder/decoder, 20 ... Security ID, 30 ... DSP, 40 ... Memory card, 42 ... Flash memory, 52 ... Security block, TRKLIST.MSF ... Track information

management file, INFLIST.MSF ... Additional information management file, A3Dnnn.MSA ... Audio data file

[Title of Document] Abstract

[Abstract]

[Subject]

In a case of writing or reading an audio and other data in the flash memory of attachable/detachable memory card, the file can be recovered whenever the file management system Fat is destroyed.

[Solving means]

Flash memory is attachable/detachable to the recorder as a memory card. The reproduction management file, FAT, and audio data file are recorded to the flash memory. A fixed length of the reproduction management file is 1 block (=16KB). An attribute header is added to the data file by each truck (music). The information to manage the whole audio data that records in the flash memory is recorded on the reproduction management file. The information for managing each track is recorded on the attribute header. BLKID for detecting the kind of block, REVISION for indicating the number of rewriting and so forth are described on these management files. Important parameters in headers are redundantly recorded on the end of 1 block.

[Selected Drawing] Fig. 12

OIPE JC151 NOV 26 2004 PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT(S): Nobuyuki KIHARA and Teppei YOKOTA

SERIAL No.: 09/674,441 Group Art Unit: 2143

FILED: November 1, 2000 Examiner: Kyung H Shin

INVENTION: DATA PROCESSING APPARATUS, DATA PROCESSING METHOD, TERMINAL UNIT, AND TRANSMISSION METHOD OF DATA PROCESSING APPARATUS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

CERTIFIED TRANSLATION

Yuka NAKAMURA residing at c/o SUGIURA PATENT OFFICE, 7th floor, Ikebukuro Park Bldg., 49-7, Minami Ikebukuro 2-chome, Toshima-ku, Tokyo, JAPAN, declares:

(1) that she knows well both the Japanese and English languages;

(2) that she translated Japanese Application No. 11-092699 from Japanese to English;

(3) that the attached English translation is a true and correct translation of the above-identified Japanese Application to the best of her knowledge and belief; and

(4) that all statements made of her own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements are made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 USC 1001, and that such false statements may jeopardize the validity of the application or any patent issuing thereon.

November 17, 2004 Yuka Nakamura

Date Yuka NAKAMURA

[Title of Document] Specification

[Title of the Invention] Data Processing Apparatus and Method

[Scope of Claims for a Patent]

[Claim 1]

A data processing apparatus for moving contents from an attachable/detachable non-volatile storing medium to a storing device,

wherein contents stored in the storing medium are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the storing medium,

wherein when the contents are moved to the storing device, the first key is decrypted by the second key and the decrypted first key is encrypted by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

wherein the first key encrypted by the third key and the encrypted contents are sent to the storing medium.

[Claim 2]

The data processing apparatus as set forth in claim 1,

wherein the storing device that has received the first key encrypted by the third key and the encrypted contents decrypts the first key by the third

key,

encrypts the decrypted first key by a fourth key unique to the storing device, and

stores the first key encrypted by the fourth key.

[Claim 3]

The data processing apparatus as set forth in claim 1,

wherein the contents are managed in the unit of a file, the first key being created for each file, the first key encrypted by the third key being sent as file management information.

[Claim 4]

The data processing apparatus as set forth in claim 3,

wherein the file management information contains information for identifying a file.

[Claim 5]

A data processing apparatus for moving contents from a storing device to an attachable/detachable non-volatile storing medium,

wherein contents stored in the storing device are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the storing device,

wherein when the contents are moved to the storing medium, the first key is decrypted by the

second key and the decrypted first key is encrypted by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

wherein the first key encrypted by the third key and the encrypted contents are sent to the storing medium.

[Claim 6]

The data processing apparatus as set forth in claim 5,

wherein the storing medium that has stored the first key encrypted by the third key and the encrypted contents decrypts the first key by the third key,

encrypts the decrypted first key by a fourth key unique to the storing medium, and

stores the first key encrypted by the fourth key.

[Claim 7]

The data processing apparatus as set forth in claim 5,

wherein the contents are managed in the unit of a file, the first key being created for each file, the first key encrypted by the third key being sent as file management information.

[Claim 8]

The data processing apparatus as set forth in

claim 7,

wherein the file management information contains information for identifying a file.

[Claim 9]

The data processing apparatus as set forth in claim 5,

wherein move history information of the contents is stored to a non-volatile memory different from the storing medium that stores the encrypted contents.

[Claim 10]

The data processing apparatus as set forth in claim 9,

wherein when the contents are moved, if the move history information of the contents is present, with reference thereto, the contents are prohibited from being moved.

[Claim 11]

The data processing apparatus as set forth in claim 5,

wherein the storing device further encrypts contents that are input from the outside, encrypts a first key for encrypting the contents by a second key unique to the storing device, and stores the encrypted contents and the encrypted key.

[Claim 12]

The data processing apparatus as set forth in

claim 5,

wherein the storing device further interfaces with a terminal unit of an electronic contents delivering system.

[Claim 13]

A data processing method for moving contents from an attachable/detachable non-volatile storing medium to a storing device,

wherein contents stored in the storing medium are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the storing medium, the method comprising the steps of:

when the contents are moved to the storing device, decrypting the first key by the second key and encrypting the decrypted first key by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

sending the first key encrypted by the third key and the encrypted contents to the storing medium.

[Claim 14]

A data processing method for moving contents from a storing device to an attachable/detachable non-volatile storing medium,

wherein contents stored in the storing device are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the

storing device, the method comprising the steps of:

when the contents are moved to the storing medium, decrypting the first key by the second key and encrypting the decrypted first key by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

sending the first key encrypted by the third key and the encrypted contents to the storing medium.

[Claim 15]

The data processing method as set forth in claim 14,

wherein move history information of the contents is stored to a non-volatile memory different from the storing medium that stores the encrypted contents.

[Claim 16]

The data processing method as set forth in claim 14,

wherein when the contents are moved, if the move history information of the contents is present, with reference thereto, the contents are prohibited from being moved.

[Detailed Description of the Invention]

[0001]

[Technical Field to which the Invention belongs]

The present invention relates to a data

processing method and a data processing method for applying a record/reproduction of an audio data by using an attachable/detachable memory card for example, devices.

[0002]

[Prior Art]

EEPROM (Electrically Erasable Programmable ROM) that is an electrically rewritable non-volatile memory requires a large space because each bit is composed of two transistors. Thus, the integration of EEPROM is restricted. To solve this problem, a flash memory that allows one bit to be accomplished with one transistor using all-bit-erase system has been developed. The flash memory is being expected as a successor of conventional record mediums such as magnetic disks and optical discs.

[0003]

A memory card using a flash memory is also known. The memory card can be freely attached to an apparatus and detached therefrom. A digital audio recording/reproducing apparatus that uses a memory card instead of a conventional CD (Compact Disc) or MD (Mini Disc) can be accomplished.

[0004]

Since an audio recorder using a memory card as a record medium records and reproduces digital data, when the audio recorder uses a compressing method that

allows data to be reproduced with relatively high quality, the copyright of music data and so forth that are recorded and reproduced should be protected. As an example, using encrypting technologies, memory cards that are not authenticated can be prohibited from being used. In other words, in a combination of an authenticated recorder and an authenticated memory card, encrypted data is decrypted.

[0005]

Conventional memory cards do not have an encrypting function. Thus, to record confidential data to a memory card, the recorder should encrypt data and record the encrypted data to a memory card. However, when a decrypting key is stored in a memory card, the security of data cannot be protected. On the other hand, when a decrypting key is stored in the recorder, encrypted data cannot be decrypted by other than the recorder that has encrypted the data. Thus, the compatibility of the memory cards cannot be maintained. For example, data stored in a memory card of one user cannot be decrypted by a recorder of another user. To solve this problem, a system of which both a recorder and a memory card have respective encrypting functions and they authenticate each other for assuring the security of data and the compatibility of the cards has been proposed.

[0006]

On the other hand, as digital audio/video information and multimedia systems are becoming common, a music data delivering service of which music data is delivered from a music data delivering server to a personal computer through a network such as Internet or a digital broadcast is being accomplished. In such a service, delivered contents data is stored in a hard disk of the personal computer.

[0007]

In a system of which a hard disk is used as an audio server, audio contents data is moved from the hard disk to a memory card. With the memory card, the moved data can be reproduced by for example a portable player. In contrast, audio data is moved from the memory card to the hard disk of the personal computer. In this case, data is moved from the hard disk to a memory card so that the data is not left in the hard disk.

[0008]

In the system of which a hard disk is used as an audio server, when data is moved from a memory card to the hard disk, all contents data of the memory card is moved to the hard disk. In this method, since an encrypting process or the like is not required, the structure becomes simple and the data can be moved at high speed. In addition, since the hard disk cannot decrypt stored data, from a viewpoint of the copyright

owner, this method is the most safety method.

[0009]

[Problem to be solved by the Invention]

However, in the case that a contents key stored in the memory card is encrypted with a storage key of the memory card, when the contents key stored in the hard disk is returned to the memory card, other than the original memory card cannot decrypt encrypted data. In other words, even if contents data is moved from the hard disk to another memory card, the contents data cannot be reproduced. In addition, when the original memory card is lost or destroyed, all data stored therein cannot be used.

[0010]

An object of the present invention is to solve a problem with respect to a data moving operation and provide a data processing apparatus and method that allow a storage key to be used for a storing unit such as a hard disk and to be re-keyed.

[0011]

Another object of the present invention is to provide a data processing apparatus and a data processing method for substantially preventing all contents data of a hard disk from being copied to many memory cards.

[0012]

[Means for Solving the Problem]

To solve the above-mentioned problem, according to the present invention of claim 1, there is provided a data processing apparatus for moving contents from an attachable/detachable non-volatile storing medium to a storing device,

wherein contents stored in the storing medium are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the storing medium,

wherein when the contents are moved to the storing device, the first key is decrypted by the second key and the decrypted first key is encrypted by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

wherein the first key encrypted by the third key and the encrypted contents are sent to the storing medium.

[0013]

According to the present invention of claim 5, there is provided a data processing apparatus for moving contents from a storing device to an attachable/detachable non-volatile storing medium,

wherein contents stored in the storing device are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the storing device,

wherein when the contents are moved to the storing medium, the first key is decrypted by the second key and the decrypted first key is encrypted by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

wherein the first key encrypted by the third key and the encrypted contents are sent to the storing medium.

[0014]

To securely prevent contents from being illegally copied, storing move history information of the contents in a non-volatile memory is recommended rather than the record medium for storing encrypted contents. In addition, when the move of contents is processed, the move of contents is prohibited by referring move history information of the contents, if move history information of the contents resides in a non-volatile memory.

[0015]

According to the present invention of claim 13, there is provided a data processing method for moving contents from an attachable/detachable non-volatile storing medium to a storing device,

wherein contents stored in the storing medium are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the

storing medium, the method comprising the steps of:

when the contents are moved to the storing device, decrypting the first key by the second key and encrypting the decrypted first key by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

sending the first key encrypted by the third key and the encrypted contents to the storing medium.

[0016]

According to the present invention of claim 14, there is provided a data processing method for moving contents from a storing device to an attachable/detachable non-volatile storing medium,

wherein contents stored in the storing device are encrypted and a first key for encrypting the contents is encrypted by a second key unique to the storing device, the method comprising the steps of:

when the contents are moved to the storing medium, decrypting the first key by the second key and encrypting the decrypted first key by a third key, the third key being shared between the storing device and the storing medium when they have been successfully authenticated, and

sending the first key encrypted by the third key and the encrypted contents to the storing medium.

[0017]

This invention is to provide a storing apparatus of a hard disk or the like having a unique key for the apparatus. When an encrypted contents at the first key are moved from the record medium such as the memory card, the first key for encrypting the contents re-encrypts by a unique key of the apparatus with changing the first key into a unique key for the record medium. That is the process of a key lock is repeated. Accordingly the repetition, whenever the contents are moved to the other record medium from the original record medium in a storing apparatus that can be encrypted.

[0018]

Besides, according to an embodiment of the present invention, move history information of the contents for the record medium is stored in the other non-volatile memory instead of a non-volatile memory of a storing apparatus that can be securely prevented the contents from being illegally copied. In other words, if move history information of the contents for the record medium isn't stored in a non-volatile memory, a copied hard disk group is prepared by executing a physical entire copy of a storing apparatus such as a hard disk. Secondly, by the replacement of a moved hard disk to a copied hard disk group before the movement, it is allowing to copy a real content toward a majority of memory card. In the process of the move,

the re-movement of a moved content is prohibited by referring move history information of the contents that prevents securely the contents from being illegally copied.

[0019]

[Embodiment of the Invention]

Next, an embodiment of the present invention will be described. Fig. 1 is a block diagram showing the structure of a digital audio signal recorder (recorder/player) using a memory card according to an embodiment of the present invention. The digital audio recorder records and reproduces a digital audio signal using a detachable memory card. For example, the recorder composes an audio system along with an amplifying unit, a speaker, a CD player, an MD recorder, a tuner, and so forth. However, it should be noted that the present invention can be applied to other audio recorders. In other words, the present invention can be applied to a portable recorder. In addition, the present invention can be applied to a recorder that records a digital audio data that is circulated as a satellite data communication, a digital broadcast, or Internet. Moreover, the present invention can be applied to a system that records/reproduces moving picture data and still picture data rather than audio data. The system according to the embodiment of the present invention

can record and reproduce additional information such as picture and text other than a digital audio signal.

[0020]

The recorder has an audio encoder/decoder IC 10, a security IC 20, a DSP (Digital Signal Processor) 30. Each of these devices is composed of a one-chip IC. A detachable memory card 40 is an attachable/detachable memory card. The one-chip IC of the memory card 40 has flash memory (nonvolatile memory), a memory control block, and a security block. The security block has a DES (Data Encryption Standard) encrypting circuit. According to the embodiment, the recording/reproducing apparatus may use a microcomputer instead of the DSP 30.

[0021]

The audio encoder/decoder IC 10 has an audio interface 11 and an encoder/decoder block 12. The encoder/decoder block 12 encodes a digital audio data corresponding to a highly efficient encoding method and writes the encoded data to the memory card 40. In addition, the encoder/decoder block 12 decodes encoded data that is read from the memory card 40. As the highly efficient encoding method, the format (it is referred to as ATRAC3) that is a modification of the ATRAC (Adaptive Transform Acoustic Coding) format used in Mini-Disc can be used.

[0022]

In the ATRAC3 format, audio data sampled at 44.1 kHz and quantized with 16 bits is highly efficiently encoded. In the ATRAC3 format, the minimum data unit of audio data that is processed is a sound unit (SU). 1 SU is data of which data of 1024 samples (1024 x 16 bits x 2 channels) is compressed to data of several hundred bytes. The duration of 1 SU is around 23 msec. In the highly efficient encoding method, the data amount of audio data is compressed to data that is around 10 times smaller than that of original data. As with the ATRAC1 format used in Mini-Disc, the audio signal is processed by the ATRAC3 format that is less deteriorating in the audio quality by the decompression and decompression corresponding to the ATRAC3 format.

[0023]

A line input selector 13 selectively supplies the reproduction output signal of an MD, the output signal of a tuner, or a reproduction output signal of a tape to an A/D converter 14. The A/D converter 14 converts the input line signal to a digital audio signal (sampling frequency = 44.1 kHz; the number of quantizing bits = 16). A digital input selector 16 selectively supplies a digital output signal of an MD, a CD, or a CS (Satellite Digital Broadcast) to a digital input receiver 17. The digital input signal is transmitted through for example an optical cable. An output signal of the digital input receiver 17 is

supplied to a sampling rate converter 15. The sampling rate converter 15 converts the digital input signal into a digital audio signal (sampling frequency = 44.1 kHz; the number of quantizing bits = 16).

[0024]

The encoder/decoder block 12 of the audio encoder/decoder IC 10 supplies encoded data to a DES encrypting circuit 22 through an interface 21 of the security IC 20. The DES encrypting circuit 22 has a FIFO 23. The DES encrypting circuit 22 is disposed so as to protect the copyright of contents. The memory card 40 also has a DES encrypting circuit. The DES encrypting circuit 22 of the recording/reproducing apparatus has a plurality of master keys and an apparatus-unique storage key. The DES encrypting circuit 22 also has a random number generating circuit. The DES encrypting circuit 22 can share an authenticating process and a session key with the memory card 40 that has the DES encrypting circuit. In addition, the DES encrypting circuit 22 can re-encrypt data with the storage key of the DES encrypting circuit.

[0025]

The encrypted audio data that is output from the DES encrypting circuit 22 is supplied to a DSP 30. The DSP 30 communicates with the memory card 40 which is attached to an attaching/detaching mechanism (not

shown) of the data processing apparatus via an interface. The DSP 30 writes the encrypted data to the flash memory of the memory card 40. The encrypted data is serially transmitted between the DSP 30 and the memory card 40. In addition, an external SRAM (Static Random Access Memory) 31 is connected to the DSP 30.

[0026]

A bus interface 32 is connected to the DSP 30. Data is supplied from an external controller (not shown) to the DSP 30 through a bus 33. The external controller controls all operations of the audio system. The external controller supplies data such as a record command or a reproduction command that is generated corresponding to a user's operation through an operation portion to the DSP 30 through the bus interface 32. In addition, the external controller supplies additional information such as image information and character information to the DSP 30 through the bus interface 32. The bus 33 is a bidirectional communication path. Additional information that is read from the memory card 40 is supplied to the external controller through the DSP 30, the bus interface 32, and the bus 33. In reality, the external controller is disposed in for example an amplifying unit of the audio system. In addition, the external controller causes a display portion to display additional information, the operation state of the

recorder, and so forth. The display portion is shared by the audio system. Since data that is exchanged through the bus 33 is not copyright protected data, it is not encrypted.

[0027]

The encrypted audio data that is read from the memory card 40 by the DSP 30 is decrypted by the security IC 20. The audio encoder/decoder IC 10 decodes the encoded data corresponding to the ATRAC3 format. Output data of the audio encoder/decoder 10 is supplied to a D/A converter 18. The D/A converter 18 converts the output data of the audio encoder/decoder 10 into an analog signal. The analog audio signal is supplied to a line output terminal 19.

[0028]

The analog audio signal is supplied to an amplifying unit (not shown) through the line output terminal 19. The analog audio signal is reproduced from a speaker or a head set. The external controller supplies a muting signal to the D/A converter 18. When the muting signal represents a mute-on state, the external controller prohibits the audio signal from being output from the line output terminal 19.

[0029]

Fig. 2 is a block diagram showing the internal structure of the DSP 30. Referring to Fig. 2, the DSP 30 comprises a core 34, a flash memory 35, an

SRAM 36, a bus interface 37, a memory card interface 38, and inter-bus bridges. The DSP 30 has the same function as a microcomputer. The core 34 is equivalent to a CPU. The flash memory 35 stores a program that causes the DSP 30 to perform predetermined processes. The SRAM 36 and the external SRAM 31 are used as a RAM of the recording/reproducing apparatus.

[0030]

The DSP 30 controls a writing process for writing encrypted audio data and additional information to the memory card 40 corresponding to an operation signal such as a record command received through the bus interfaces 32 and 37 and a reading process for reading them therefrom. In other words, the DSP 30 is disposed between the application software side of the audio system that records/reproduces audio data and additional information and the memory card 40. The DSP 30 is operated when the memory card 40 is accessed. In addition, the DSP 30 is operated corresponding to software such as a file system.

[0031]

The DSP 30 manages files stored in the memory card 40 with the FAT system used in conventional personal computers. In addition to the file system, according to the embodiment of the present invention, a track information management file is used. The track information management file will be descried later.

The track information management file as the first file management information is used to manage audio data files. On the other hand, the FAT as the second file management information is used to mange all files including audio data files and track information management files stored in the flash memory of the memory card 40. The track information management file is stored in the memory card 40. The FAT is written to the flash memory along with the route directory and so forth before the memory card 40 is shipped.

[0032]

According to the embodiment of the present invention, to protect the copyright of data, audio data that has been compressed corresponding to the ATRAC3 format is encrypted. On the other hand, since it is not necessary to protect the copyright of the an additional information and a track information management file, it is not encrypted. There are two types of memory cards that are an encryption type and a non-encryption type. However, a memory card for use with the recorder/player that records copyright protected data is limited to the encryption type.

[0033]

Fig. 3 is a block diagram showing the internal structure of the memory card 40. The memory card 40 comprises a control block 41 and a flash memory 42 that are structured as a one-chip IC. A

bidirectional serial interface is disposed between the DSP 30 of the recorder/player and the memory card 40. The bidirectional serial interface is composed of ten lines that are a clock line SCK for transmitting a clock signal that is transmitted along with data, a status line SBS for transmitting a signal that represents a status, a data line DIO for transmitting data, an interrupt line INT, two GND lines, two INT lines, and two reserved lines.

[0034]

The clock line SCK is used for transmitting a clock signal in synchronization with data. The status line SBS is used for transmitting a signal that represents the status of the memory card 40. The data line DIO is used for inputting and outputting a command and encrypted audio data. The interrupt line INT is used for transmitting an interrupt signal that causes the memory card 40 to interrupt the DSP 30 of the recorder/player. When the memory card 40 is attached to the recorder/player, the memory card 40 generates the interrupt signal. However, according to the embodiment of the present invention, since the interrupt signal is transmitted through the data line DIO, the interrupt line INT is grounded.

[0035]

A serial/parallel converting, parallel/serial converting, and interface block (S/P, P/S, I/F block)

43 is an interface disposed between the DSP 30 of the recorder/player and the control block 41 of the memory card 40. The S/P, P/S, and IF block 43 converts serial data received from the DSP 30 of the recorder/player into parallel data and supplies the parallel data to the control block 41. In addition, the S/P, P/S, and IF block 43 converts parallel data received from the control block 41 into serial data and supplies the serial data to the DSP 30. When the S/P, P/S, and IF block 43 receives a command and data through the data line DIO, the S/P, P/S, and IF block 43 separates them into these that are normally accessed to the flash memory 42 and those that are encrypted.

[0036]

In the format of which data is transmitted through the data line DIO, after a command is transmitted, data is transmitted. The S/P, P/S, and IF block 43 detects the code of a command and determines whether the command and data are those that are normally accessed or those that are encoded. Corresponding to the determined result, the S/P, P/S, and IF block 43 stores a command that is normally accessed to a command register 44 and stores data that is normally accessed to a page buffer 45 and a write register 46. In association with the write register 46, the memory card 40 has an error correction code encoding circuit 47. The error correction code

encoding circuit 47 generates a redundant code that is an error correction code for data temporarily stored in the page buffer 45.

[0037]

Output data of the command register 44, the page buffer 45, the write register 46, and the error correction code encoding circuit 47 is supplied to a flash memory interface and sequencer (hereinafter, referred to as memory I/F and sequencer) 51. The memory IF and sequencer 51 is an interface disposed between the control block 41 and the flash memory 42 and controls data exchanged therebetween. Data is written to the flash memory 42 through the memory IF and sequencer 51.

[0038]

By the security IC 20 of the recorder and the security block 52 of the memory card 42, contents are written to the flash memory 42 (audio data that has been compressed corresponding to the ATRAC3 format, hereinafter, this audio data is referred to as ATRAC3 data) is encrypted to protect the copyright of the ATRAC3 data. The security block 52 comprises a buffer memory 53, a DES encrypting circuit 54, and a nonvolatile memory 55.

[0039]

The security block 52 of the memory card 40 has a plurality of authentication keys and a unique

storage key for each memory card. The nonvolatile memory 55 stores a key necessary for encrypting data which is unseen from the outer side. For example, a storage key is stored in the nonvolatile memory 55. The security block 52 also has a random number generating circuit. The security block 52 authenticates a specified recorder (it means that using a data format and/or the like are in the same system) and shares a session key therewith. In addition, the security block 52 can be re-encrypted contents with the storage key through the DSE encrypting circuit 54.

[0040]

For example, when the memory card 40 is attached to the recorder, they are mutually authenticated. The security IC 20 of the recorder and the security block 52 of the memory card 40 mutually authenticate. When the recorder has authenticated the attached memory card 40 (a memory card is in the same system) and the memory card 40 has authenticated the recorder (a record is in the same system), they are mutually authenticated. After the mutual authenticating process has been successfully performed, the recorder and the memory card 40 generate respective session keys and share them with each other. Whenever the recorder and the memory card 40 authenticate each other, they generate respective session keys.

[0041]

When contents are written to the memory card 40, the recorder encrypts a contents key with a session key and supplies the encrypted data to the memory card 40. The memory card 40 decrypts the contents key with the session key, re-encrypts the contents key with a storage key, and supplies the contents key to the recorder. The storage key is a unique key for each memory card 40. When the recorder receives the encrypted contents key, the recorder performs a formatting process for the encrypted contents key, and writes the encrypted contents key and the encrypted contents to the memory card 40.

[0042]

Data that is read from the flash memory 42 is supplied to the page buffer 45, the read register 48, and the error correction circuit 49 through the memory IF and the sequencer 51. The error correcting circuit 49 corrects an error of the data stored in the page buffer 45. Output data of the page buffer 45 that has been error-corrected and the output data of the read register 48 are supplied to the S/P, P/S, and IF block 43. The output data of the S/P, P/S, and IF block 43 is supplied to the DSP 30 of the recorder through the above-described serial interface.

[0043]

When data is read from the memory card 40,

the contents key encrypted with the storage key and the contents encrypted with the block key are read from the flash memory 42. The security block 52 decrypts the contents key with the storage key. The security block 52 re-encrypts the decrypted content key with the session key and transmits the re-encrypted contents key to the recorder. The recorder decrypts the contents key with the received session key and generates a block key with the decrypted contents key. The recorder successively decrypts the encrypted ATRAC3 data.

[0044]

A config. ROM 50 is a configuration ROM that stores partition information, various types of attribute information, and so forth of the memory card 40. The memory card 40 also has an erase protection switch 60. When the switch 60 is in the erase protection position, even if a command that causes the memory card 40 to erase data stored in the flash memory 42 is supplied from the recorder side to the memory card 40, the memory card 40 is prohibited from erasing the data stored in the flash memory 42. An OSC cont. 61 is an oscillator that generates a clock signal that is the reference of the timing of the process of the memory card 40.

[0045]

Fig. 4 is a schematic diagram showing the hierarchy of the processes of the file system of the

computer system that uses a memory card as a storage medium. On the hierarchy, the top hierarchical level is an application process layer. The application process layer is followed by a file management process layer, a logical address management layer, a physical address management layer, and a flash memory access layer. In the hierarchical structure, the file management process layer is the FAT file system. Physical addresses are assigned to individual blocks of the flash memory. The relation between the blocks of the flash memory and the physical addresses thereof does not vary. Logical addresses are addresses that are logically handled on the file management process layer.

[0046]

Fig. 5 is a schematic diagram showing the physical structure of data handled in the flash memory 42 of the memory card 40. In the memory 42, a data unit (referred to as segment) is divided into a predetermined number of blocks (fixed length). One block is divided into a predetermined number of pages (fixed length). In the flash memory, data is erased as each block at a time. Data is written to the flash memory 42 or read therefrom as a page at a time. The size of each block is the same. Likewise, the size of each page is the same. One block is composed of page 0 to page m. For example, one block has a storage

capacity of for example 8 KB (kilobytes) or 16 KB. One page has a storage capacity of 512 B (bytes). When one block has a storage capacity of 8 KB, the total storage capacity of the flash memory 42 is 4 MB (512 blocks) or 8 MB (1024 blocks). When one block has a storage capacity of 16 KB, the total storage capacity of the flash memory 42 is 16 MB (1024 blocks), 32 MB (2048 blocks), or 64 MB (4096 blocks).

[0047]

One page is composed of a data portion of 512 bytes and a redundant portion of 16 bytes. The first three bytes of the redundant portion is an overwrite portion that is rewritten whenever data is updated. The first three bytes successively contain a block status area, a page status area, and an update status area. The remaining 13 bytes of the redundant portion are fixed data that depends on the contents of the data portion. The 13 bytes contain a management flag area (1 byte), a logical address area (2 bytes), a format reserve area (5 bytes), a dispersion information ECC area (2 bytes), and a data ECC area (3 bytes). The dispersion information ECC area contains redundant data for an error correction process against the management flag area, the logical address area, and the format reserve area. The data ECC area contains redundant data for an error correction process against 512-byte data.

[0048]

The management flag area contains a system flag (1: user block, 0: boot block), a conversion table flag (1: invalid, 0: table block), a copy prohibition flag (1: OK, 0: NG), and an access permission flag (1: free, 0: read protect).

[0049]

The first two blocks - blocks 0 and 1 are boot blocks. The block 1 is a backup of the block 0. The boot blocks are top blocks that are valid in the memory card. When the memory card is attached to the recorder, the boot blocks are accessed at first. The remaining blocks are user blocks. Page 0 of the boot block contains a header area, a system entry area, and a boot and attribute information area. Page 1 of the boot block contains a prohibited block data area. Page 2 of the boot block contains a CIS (Card Information Structure)/IDI (identify Drive Information) area.

[0050]

The header area of the boot block contains a boot block ID and the number of effective entries. The system entries are the start position of prohibited block data, the data size thereof, the data type thereof, the data start position of the CIS/IDI area, the data size thereof, and the data type thereof. The boot and attribute information contains the memory card type (read only type, rewritable type, or hybrid type),

the block size, the number of blocks, the number of total blocks, the security/non-security type, the card fabrication data (date of fabrication), and so forth.

[0051]

Since the flash memory has a restriction for the number of rewrite times due to the deterioration of the insulation film, it is necessary to prevent the same storage area (block) from being concentratedly accessed. Thus, when data at a particular logical address stored at a particular physical address is rewritten, updated data of a particular block is written to a non-used block rather than the original block. Thus, after data is updated, the relation between the logical address and the physical address changes. Consequently, the same block is prevented from being concentratedly accessed by executing this process (this process is referred to as swap process). Thus, the service life of the flash memory can be prolonged.

[0052]

The logical address associates with data written to the block. Even if the block of the original data is different from the block of updated data, the address on the FAT does not change. Thus, the same data can be properly accessed. However, since the swap process is performed, a conversion table that correlates logical addresses and physical addresses is

required (this table is referred to as logical-physical address conversion table). With reference to the logical-physical address conversion table, a physical address corresponding to a logical address designated on the FAT is obtained. Thus, a block designated with a physical address can be accessed.

[0053]

The DSP 30 stores the logical-physical address conversion table in the SRAM. When the storage capacity of the RAM is small, the logical-physical address conversion table can be stored to the flash memory. The logical-physical address conversion table correlates logical addresses (2 bytes) sorted in the ascending order with physical addresses (2 bytes). Since the maximum storage capacity of the flash memory is 128 MB (8192 blocks), 8192 addresses can be assigned with two bytes. The logical-physical address conversion table is managed for each segment. Thus, the size of the logical-physical address conversion table is proportional to the storage capacity of the flash memory. When the storage capacity of the flash memory is 8 MB (two segments), two pages are used as the logical-physical address conversion table for each of the segments. When the conversion table is stored in the flash memory, a predetermined one bit of the management flag area in the redundant portion in each page represents whether or not the current block is a

block containing the logical-physical address conversion table.

[0054]

The above-described memory card can be used with the FAT file system of a personal computer system as with the disc shaped record medium. The flash memory has an IPL area, a FAT area, and a route directory area (not shown in Fig. 5). The IPL area contains the address of a program to be initially loaded to the memory of the recorder. In addition, the IPL area contains various types of memory information. The FAT area contains information with respect to blocks (clusters). The FAT has defined unused blocks, next block number, defective blocks, and last block number. The route directory area contains directory entries that are a file attribute, an update date [day, month, year], file size, and so forth.

[0055]

According to the embodiment of the present invention, in addition to the file management system defined in the format of the memory card 40, the file management information (track information management file) is provided to the music file. The track information management file is recorded to a user block of the flash memory 42 of the memory card 40. Thus, as will be described later, even if the FAT of the memory card 40 is destroyed, a file can be recovered.

[0056]

The track information management file is generated by the DSP 30. When the power of the recorder is turned on, the DSP 30 determines whether or not the memory card 40 has been attached to the recorder. When the memory card has been attached, the DSP 30 authenticates the memory card 40. When the DSP 30 has successfully authenticated the memory card 40, the DSP 30 reads the boot block of the flash memory 42. Thus, the DSP 30 reads the physical-logical address conversion table and stores the read data to the SRAM. The FAT and the route directory have been written to the flash memory of the memory card 40 before the memory card 40 is shipped. When data is recorded to the memory card 40, the track information management file is generated.

[0057]

In other words, a record command issued by the remote controller of the user or the like is supplied to the DSP 30 from the external controller through the bus and the bus interface 32. The encoder/decoder IC 10 compresses the received audio data and supplies the resultant ATRAC3 data to the security IC 20. The security IC 20 encrypts the ATRAC3 data. The encrypted ATRAC3 data is recorded to the flash memory 42 of the memory card 40. Thereafter, the FAT and the track information management file are

updated. Whenever a file is updated (in reality, whenever the recording process of audio data is completed), the FAT and the track information management file stored in the SRAMs 31 and 36 are rewritten. When the memory card 40 is detached or the power of the recorder is turned off, the FAT and the track information management file that are finally supplied from the SRAMs 31 and 36 are recorded to the flash memory 42. Alternatively, whenever the recording process of audio data is completed, the FAT and the track information management file written in the flash memory 42 may be rewritten. When audio data is edited, the contents of the track information management file are updated.

[0058]

In the data structure according to the embodiment, additional information management file is updated and recorded to the flash memory 42. Likewise the track information management file, the additional information management file is also created and updated. The additional information is supplied from the external controller to the DSP 30 through the bus and the bus interface 32. The additional information is recorded to the flash memory 42 of the memory card 40. Since the additional information is not supplied to the security IC 20, it is not encrypted. When the memory card 40 is detached from the recorder or the

power thereof is turned off, the additional information management file is written from the SRAM of the DSP 30 to the flash memory 42. When an additional information is recorded, an additional information management file can be written on the flash memory 42.

[0059]

Fig. 6 is a schematic diagram showing the file structure of the memory card 40. As the file structure, there are a still picture directory, a moving picture directory, a voice directory, a control directory, and a music directory. According to the embodiment, music programs are recorded and reproduced. Next, the music directory will be described. The music directory consists the track information management file INFLIST.MSF, backup TRKLISTB.MSF for the track information management file, INFLIST.MSF for recording a variety of additional data such as artist name, ISRC code, timestamp, still picture data and or the like and an ATRAC3 data file A3Dnnnn.MSA. The area NAME1 is an area that records the memory card name, the program block (for one byte code corresponding to ASCII/8859-1 character code). The area NAME2 is an area that records the memory card name and the program name (for two byte code corresponding to MS-JIS/Hankul/Chinese code).

[0060]

Fig. 7 is a schematic diagram showing the

relations between the track information management file TRKLIST.MSF of the music directory, NAME1, NAME2, and ATRAC3 data file A3Dnnnn.MSA. TRKLIST.MSF is a fixed-length file of 64 KB (= 16K x 4). The first half of 32 KB is used for recording the parameter which manages the track. The second half of 32 KB is used for recording NAME 1 and NAME 2. NAME1 and NAME2 files that recorded the music program and so forth can be individually executed however, in the case of the system with a small RAM capacity can be smoothly managed and operated without separating the track information management file and the program files.

[0061]

ATRAC3 data file A3Dnnnn.MSA. and an additional information file INFLIST.MSF are managed by the track information area TRKINF-nnnn and the parts information area PRTINF-nnnn in the track information management file TRKLIST.MSF. Thus, ATRAC3 data file A3Dnnnn.MSA. is only encrypted. In Fig. 6, the data length in the horizontal direction is 16 bytes (0 to F). A hexadecimal number in the vertical direction represents the value at the beginning of the current line.

[0062]

Next, with reference to Figs. 8, the relation between music programs and ATRAC3 data files will be described. One track is equivalent to one music

program. The number of tracks which can be recorded in the memory card that is limited as maximumly 400 tracks. In addition, one music program is composed of one ATRAC3 data. The ATRAC3 data file is audio data that has been compressed corresponding to the ATRAC3 format. The ATRAC3 data file is recorded as a cluster at a time to the memory card 40. One cluster has a capacity of 16 KB. A plurality of files are not contained in one cluster. The minimum data erase unit of the flash memory 42 is one block. In the case of the memory card 40 for music data, a block is a synonym of a cluster. In addition, one cluster is equivalent to one sector.

[0063]

One music program is basically composed of one part. However, when a music program is edited, one music program may be composed of a plurality of parts. The connection of parts of a music program is recorded in the track information management file TRKLIST.MSF. The parts is the data unit that represents a recorded data from the beginning to the end of continuing recording time. Normally, one track is composed of one part. There is the limitation with the maximum value of the parts. When the number of parts is decided as P and the number of tracks are set to T (=1 to 400), there is the relation (P =2043-4xT) between the parts and tracks which are available for the use. For

example, if one track is composed of 2039 parts. No parts can be allocated to a second program that allows impossibly making a second program file.

[0064]

The minimum unit of a part is the sound unit (it is referred as SU). In addition, SU is the minimum data unit in the case that audio data is compressed corresponding to the ATRAC3 format. 1 SU is audio data of which data of 1024 samples at 44.1 kHz (1024 x 16 bits x 2 channels) is compressed to data that is around 10 times smaller than that of original data. The duration of 1 SU is around 23 msec. Normally, one part is composed of several thousand SU.

[0065]

Fig. 8 is a schematic diagram showing the file structure in the case that two music programs of a CD or the like are successively recorded. The first program (file 1) is composed of for example five clusters. Since one cluster cannot contain two files of the first program and the second program, the file 2 starts from the beginning of the next cluster. Thus, the end of the part 1 corresponding to the file 1 is in the middle of one cluster and the remaining area of the cluster contains no data. Likewise, the second music program (file 2) is composed of one part.

[0066]

There are four types of edit processes that

are a divide process, a combine process, an erase process, and a move process. The divide process is performed to divide one track into two portions. When the divide process is performed, the number of total tracks increases by one. In the divide process, one file is divided into two files on the file system. Thus, in this case, the track information management file TRKLIST.MSF and the FAT are updated. The erase process is performed to erase a track. The track numbers after the track that has been erased decrease one by one. The move process is performed to change the track sequence. Thus, when the erase process or the move process is performed, the reproduction track information management file and the FAT are updated.

[0067]

Fig. 8 is a schematic diagram showing the combined result of two programs (file 1 and file 2) shown in Fig. 9. As a result of the combine process, the combined file is composed of two parts. Fig. 10 is a schematic diagram showing the divided result of which one program (file 1) is divided in the middle of the cluster 2. By the divide process, the file 1 is composed of clusters 0, 1, and the beginning portion of cluster 2. The file 2 is composed of the end portion of cluster 2 and clusters 3 and 4.

[0068]

In the case of the above-described editing

process is performed, the block of the file management information TRKLIST.MSF which contains editing points is only rewritten because rewriting an ATRAC3 data file takes much longer. Consequently, the concept of the part is introduced.

[0069]

Fig. 11 shows the detailed data structure of the track information management file TRKLIST.MSF. The size of the track information management file TRKLIST.MSF is one cluster (one block = 16 KB). In the track information management file, the first 32 bytes of (0x0000) to (0x0010) shown in Fig. 11 is used for the header. In the file, 8-byte areas are referred to as slots. In the case of the track information management file, 16-byte areas are referred to as slots. In order, the data is supplied to those headers from the first header.

[0070]

= BLKID-TL0 (4 bytes)

Value: Fixed value = (TL0 = 0x544C2D30, TL1 = 0x544C2D31)

T-TRK (2 bytes)

Represents about the number of total tracks (1 to 400)

Mcode (2 bytes)

Represents code to identify the maker and model of the recorder/player.

Mcode is management code to identify the maker of recorder that records some information to memory card. Mcode is given when licenced. The type code is managed by makers individually that are licenced.

REVISION (4 bytes)

Represents the number of TRKLIST rewrite times, REVISION is incremented in terms of recording.

YMDhms (4 bytes)

Represents update date and time data.

N1 (OP)(1 byte)

Represents the sequential number of the memory card (numerator side). When one memory card is used, the value of the area N1 is 0x01. "OP" means optional term.

N2 (OP)(1 byte)

Represents the sequential number of the memory card (denominator side). When one memory card is used, the value of the area N2 is 0x01.

MSID (OP)(2 bytes)

Represents the ID of a memory card. When a plurality of memory cards is used, the value of the area MSID of each memory card is the same (T.B.D.). (T.B.D. (to be defined) represents that this value may be defined in future).

S-TRK (2 bytes).

Represents a special track (T.B.D.). Normally, the value of the area S-TRK is 0x0000.

PASS (OP)(2 bytes)

Represents a password (T.B.D.).

APP (OP)(2 bytes)

Represents the definition of a reproduction application (T.B.D.) (normally, the value of the area APP is 0x0000).

INF-S (OP)(2 bytes)

Represents the additional information pointer of the entire memory card. When there is no additional information, the value of the area INF-S is 0x00.

S_YTMDhms (OP)(4 bytes)

Represents TRKLIST.MSF.'s update date and time.

[0071]

The last 16 bytes of the file TRKLIST are used for an area BLKID-TL0, an area Mcode, and an area REVISION that are the same as those of the header. The backup file TRKLISTB contains the above-described header. In this case, the header contains an area BLKID-TL1, an area Mcode, and an area REVISION.

[0072]

While data is being recorded to a memory card, it may be mistakenly or accidentally detached or the power of the recorder may be turned off. When such an improper operation is performed, a defect should be

detected. As described above, the REVISION area is placed at the beginning and end of each block. Whenever data is rewritten, the value of the REVISION area is incremented. If a defect termination takes place in the middle of a block, the value of the REVISION area at the beginning of the block does not match the value of the REVISION area at the end of the block. Thus, such a defect termination can be detected. Since there are two REVISION areas, the abnormal termination can be detected with a high probability. When an abnormal termination is detected, an alarm such as an error message is generated.

[0073]

In addition, since the fixed value BLKID-TL0 is written at the beginning of one block (16 KB), when the FAT is destroyed, the fixed value is used as a reference for recovering data. In other words, with reference to the fixed value, the type of the file can be determined. Since the fixed value BLK ID-TL0/TL1 are redundantly written at the header and the end portion of each block, the reliability can be secured.

[0074]

The data amount of an ATRAC3 data file is much larger than that of the track information track information management file. In addition, as will be described later, a block number BLOCK SERIAL is added to ATRAC3 data file. However, since a plurality of

ATRAC3 files are recorded to the memory card, to prevent them from become redundant, both CONNUM0 and BLOCK SERIAL are used. Otherwise, when the FAT is destroyed, it will be difficult to recover the file. In other words, one ATRAC3 data file may be composed of a plurality of blocks that are dispersed. To identify blocks of the same file, CONNUM0 is used. In addition, to identify the order of blocks in the ATRAC3 data file, BLOCK SERIAL is used.

[0075]

Likewise, the maker code (Mcode) is redundantly recorded at the beginning and the end of each block so as to identify the maker and the model in such a case that a file has been improperly recorded in the state that the FAT has not been destroyed.

[0076]

The header is followed by a track information area TRKINF for information with respect to each track and a part information area PRTINF for information with respect to each part of tracks (music programs). Fig. 11 shows the areas preceded by the area TRKLIST. The lower portion of the area TRKLISTB shows the detailed structure of these areas. In Fig. 11, a hatched area represents an unused area.

[0077]

= T0 (1 byte)

Fixed value (T0 = 0x74)

= INF-nnn (Option) (2 bytes)

Represents the additional information pointer (0 to 409) of each track. 00: music program without additional information.

= FNM-nnn (4 bytes)

Represents the file number (0x0000 to 0xFFFF) of an ATRK3 data file.

The number nnnn (in ASCII) of the ATRAC3 data file name (A3Dnnnn) is converted into 0xnnnnn.

= APP_CTL (4 bytes) (Option)

Represents an application parameter (T.B.D.) (Normally, the value of the area APP_CTL is 0x0000).

= P-nnn (2 bytes)

Represents the number of parts (1 to 2039) that compose a music program. This area corresponds to the above-described area T-PART.

= PR (1 byte)

Fixed value (PR = 0 x 50).

[0078]

S-SAM(D) SERIAL (16 bytes)

Represents serial number of apparatus that has recorded tha memory card.

APP_CTL (4 bytes) (Option)

Represents an application parameter (T.B.D.) (Normally, the value of the area APP_CTL is 0x0000).

CONNUM (4 bytes)

Represents a unique value cumulated for each

music program. The value is stored in the security block of the recorder, so as not to overlap in a memory card.

P-nnn (2 bytes)

Represents the number of parts (1 to 2039) that compose a music program.

XT (OP)(2 bytes)

Represents the reproduction duration (SU) from the point designated by INX.

0000: no setting; FFFF: up to end of music program

INX-nnn(OP) (2 bytes)

Represents a pointer that indicates specified part of music program. INX-nnn shows the number of relative SU.

YMDhms-S (4 bytes)

Represents reproduction end date and time of track with reproduction restriction.

When not used, the value of the YMDhms-S is 0x00000000.

YMDhms-E (4 bytes)

Represents reproduction end date and time of track with reproduction restriction.

When not used, the value of the YMDhms-E is 0x00000000.

MT (1 byte)

Represents the maximum number of permitted

reproduction times.

When not used, the value of MT is 0x00.

CT (1 byte)

Represents number of reproduction of the track to which reproduction condition is regulated.

When not used, the value of the area CT is 0x00.

CC (1 byte)

Controls the copy operation. 00: copy prohibited, 01: one time copy operation permitted, 10: unimited copy operation permitted. When one time copy operation permitted, copied track is prohibited to be copied.

CN (1 byte)

CN is concerned about number of permitted copy times. 00: Copy prohibited, 01 to 0xFE: Number of times, 0xFF: Unlimited copy times, it is valid about one time copy. Copied time is counted in terms of copy operation is done.

[0079]

In the part information area PRINTF-nnn, the part information is desucribed as illustrated below.

[0080]

PR (1 byte)

Fixed value (PR = 0 x 50).

A-nnnn (2 byte)

Represents the attribute information on

parts. A-nnnn consists of mode(1 byte) and SCMS(Serial Copy Management System) information(1 byte).

PRTSIZE-nnnn (4 bytes)

Represents the cluster size of a part(2 bytes), start SU(1 byte) , and end SU(1 byte).

PRTKEY-nnnn (8 bytes)

PRTKEY-nnnn is used with CONTENTSKEY to generates block key to encrypte music data.

PRTKEY-nnnn's initial value = 0, and PRTKEY-nnnn is incrimented +1 whenever a part generates by progression of edit operation.

[0081]

The mode information that indicates the mode of ATRAC3, and that is described by lower byte of A-nnnn information, is defined as illustrated in Fig. 12. Fig. 12 shows the number of bytes and recording period(in case 64MB), data translation rate, and compression rate about 6 kinds of mode, such as HQ, SP, CD, LP1, LP2, mono.

[0082]

Fig. 13 shows the content of information described by upper byte. Bit 0 forms information about emphasis on/off state, Bit 1 forms information about reproduction SKIP or normal reproduction, and Bit 2 forms information about data division for example audio data or data noise such as FAX or the like. Bit 3 and Bit 4 forms registration. SCMS information is

formed by combining Bit 5 and Bit 6, as shown in Fig. 13. Bit 7 forms information about write prohibition or permission.

[0083]

Fig. 14 shows the detailed structure of the area NAME1 (for one byte code area). Each of the areas NAME1 and NAME2 (that will be described later) is segmented with eight bytes. Thus, their one slot is composed of eight bytes. At 0x8000 that is the beginning of each of these areas, a header is placed. The header is followed by a pointer and a name. The last slot of the area NAME1 contains the same areas as the header.

[0084]

= BLKID-NM1 (4 bytes)

Represents the contents of a block (fixed value) (NM1 = 0x4E4D2D31).

MCode(2 bytes)

Represents code to identify the maker and model of recorder/player.

[0085]

= PNM1-nnn (4 bytes) (Option)

Represents the pointer to the area NM1 (for one byte code).

= PNM1-S

Represents the pointer to a name representing a memory card.

nnn (= 1 to 408) represents the pointer to a music program title.

The pointer represents the start position (2 bytes) of the block, the character code type (2 bits), and the data size (14 bits).

= NM1-nnn (Option)

Represents the memory card name and music program title for one byte code (variable length). An end code (0x00) is written at the end of the area.

[0086]

Fig. 15 shows the detailed data structure of the area NAME2 (for two byte code). At 0x8000 that is the beginning of the area, a header is placed. The header is followed by a pointer and a name. The last slot of the area NAME2 contains the same areas as the header.

[0087]

= BLKID-NM2 (4 bytes)

Represents the contents of a block (fixed value) (NM2 = 0x4E4D2D32).

MCode(2 bytes)

Represents code to identify the maker and model of recorder/player.

[0088]

= PNM2-nnn (4 bytes) (Option)

Represents the pointer to the area NM2 (for two byte code).

PNM2-S represents the pointer to the name representing the memory card. nnn (= 1 to 408) represents the pointer to a music program title.

The pointer represents the start position (2 bytes) of the block, the character code type (2 bits), and the data size (14 bits).

The starting point is represented by byte offset value(0x000-ox3989) from the beggining of NM1 region.

The types of letter code are (0:ASCII, 1:ASCII + kana letter, 2:revised 8859-1)

Data size(14 bits) is the value(0x000-0x398C) that is caluculated by adding letter data and terminal(0x00).

= NM2-nnn (Option)

Represents the memory card name and music program title for two byte code (variable). An end code (0x0000) is written at the end of the area.

[0089]

Fig. 16 shows the data arrangement (for one block) of the ATRAC3 data file A3Dnnnn in the case that 1 SU is composed of N bytes. In this file, one slot is composed of eight bytes. Fig. 15 shows the values of the top portion (0x0000 to 0x3FF8) of each slot. The first four slots of the file are used for a header. The following data is recorded in the header. The second last slot of the block redundantly contains an

area BLOCK SEED. The last slot contains areas BLKID-A3D and MCode.

[0090]

BLK ID-A3D (4 bytes)

fixed value(A3D=0x41324420) to designate the content of block.

MCode (2 bytes)

Represents the code to identify the maker and moderl of the recorder/player.

Mcode is needed to rewritten when edit process is performed.

BLOCK SEED (8 bytes)

BLOCK SEED is used to generate block key that is necessary for encrypting process.

The beginning value of BLOCK SEED is put by caluculation of random number performed by security block, and the value corresponding to the following block is incrimented +1.

The same value is written at the starting and the ending of the block, as a countermeasure against error.

BLOCK SEED is not needed to rewritten even when edit process is performed.

CONNUM (4 bytes)

CONNUM is the initial contents cumulation number.

Initially the value of CONNUM is the same

value as the value of CONNUM of TRKLIST.MSF.

CONNUM is not needed to rewritten even when edit process is performed.

BLOCK SERIAL (4 bytes)

BLOCK SERIAL is put to 0 to the beginning block, and incrimented +1 to following blocks.

BLOCK SERIAL is not needed to rewritten even when edit process is performed.

INITIALIZATION VECTOR (8 bytes)

INITIALIZATION VECTOR is the initial value that is necessary for encrypting/decrypting process on ATTRAC3 data.

The value of INITIALIZATION VECTOR corresponding to the beginning of contents is 0.

The value of INITIALIZATION VECTOR corresponding to the following block is the last encrypted value in the encrypted values corresponding to the last SU.

INITIALIZATION VECTOR is not needed to rewritten even when edit process is performed.

[0091]

The header is followed by the sound unit data SU-nnnn. SU is the data formed by compressing 1024 samples. The data amound of SU is varied in terms of mode. SU is not needed to rewritten even when edit process is performed. Fig. 17 shows the relationship between mode and the data amount of SU, the number of

SU per 1 block, and the amount of residual (reserved) data, transferring rate, and time.

[0092]

As one example, a case in which 64MB memory card is used , and CD mode is operated will be referred. 64MB memory card has 3968 blocks, and 1 SU corresponds to 320 bytes in CD mode. Thus 1 Block has 51 SU's. Therefore, 1block corresponds (1024/44100) x 51 x (3968-16) = 4680(sec) = 78(min). Transferring rate is

(44100/1024) x 320 x 8 = 110250 bps

[0093]

Fig. 18 shows the detailed data structure of the additional information management file INFLIST.MSN that contains additional information.

INFLIST.MSN is a part of track information management file TRKLIST.MSF, thus INFLIST.MSN is fragmented in terms of 16 byte ,starting from the beginning of file. The following header is placed at the beginning (0x0000) of the file INFLIST. The header is followed by the following pointer and areas.

[0094]

BLK ID-INF (4 bytes)

Represents the contents of the block (fixed value) (INF = 0x494E464F).

T-DAT (2 bytes)

Represents the number of total data areas (0

to 409).

Mcode (2 bytes)

Represents the code to identify the maker and model of the recorder/player

YMDhms (4 bytes)

Represents the record updated date and time.

INF-nnnn (4 bytes)

INF-nnnn is the pointer to the area DATA of the additional information (variable length, as 2 bytes (slot) at a time).

The start position is represented with the high order 16 bits (0000 to FFFF).

INF-nnnn Represents the offset value from the beginning of DataSlot-0000 (0x0800).

The data size is represented with low order 16 bits (0001 to 7FFF). A disable flag is set at the most significant bit. MSB = 0 (Enable), MSB = 1 (Disable)

The data size represents the total data amount of the music program.

(The data starts from the beginning of each slot. The non-data area of the slot is filled with 00.)

The first INF represents a pointer to additional information of the entire album (normally, INF-409).

[0095]

Fig. 19 shows the structure of additional

information. An 8-byte header described below is placed at the beginning of one additional information data area.

[0096]

IN (1 byte)

fixed value (IN=0x69)

ID (1 byte)

Represents the larger classification of additional information.

ID is called key ID compared to sub ID.

SID(1 byte)

Represents the classification of sub ID(T. B. D.)

SIZE (2 bytes)

Represents the size of each additional information in terms of slot (1 to 7FFF).

Disable flag is set on most sibnificant bit MSB. MSB=0(Enable), MSB=1(Disable)

MCode (2 bytes)

Represents the code to identify the maker and model of the recorder that has recorded.

[0097]

Fig. 20 shows a example of additional information. When size is 0x8xxx, erase or disable data is indicated. Each additional information is to be distinguished with the code in header, such as key ID and SID. However, the values of key ID and SID is

not shown, since the values of key ID and SID is not defined. The additional information includes copy right code ISRC(International Standard Recording Code), the information about music such as the name of Composer and artist, and hardware management information. The letter code that is described in the starting 2 byte of data added to music information.

[0098]

Fig. 21 shows the structure of one additional information. In reference to the structure shown in Fig. 21, some concrete example of additional information will be described. Fig. 22 shows a example in which additional information is time stamp. The time stamp is time stamp on recording process as shown in Fig. 20. The data is YMDhms, and 00 is written in residual area. Fig. 23 shows a example in which additional information is reproduction log file. Year-month-day(YMD) data and hour-minuite-second(HMS) data is written.

[0099]

Fig. 24 shows a example in which additional information is artist name + ISRC code + TOCID. In this example, artist name is described in 1 byte code. 00 is written in residual area of the slot. ISRC code is described as a data in the following slot. TOCID data is described as a data in the following slot. When the additional data shown in Fig 24 is

erased, the additional data shown in Fig 24 is rewritten to the data shown in Fig 25. In other word, SIZE is to be (8xxx).

[0100]

According to the embodiment of the present invention, in addition to the file system defined as a format of the memory card, the track information management file TRKLIST.MSF for music data is used. Thus, even if the FAT is destroyed, the file can be recovered. Fig. 26 shows a flow of a file recovering process. To recover the file, a computer (The computer has a function equivalent to the DSP30) that operates with a file recovery program and that can access the memory card and a storing device (hard disk, RAM, or the like) connected to the computer are used. In the first step 101, the operation that will be performed as the following.

[0101]

All blocks of the flash memory whose FAT has been destroyed are searched for TL-0 as the value (BLKID) at the top position of each block. In addition, all the blocks are searched for NM-1 as the value (BLKID) at the top position of each block. Thereafter, all the blocks are searched for NM-2 as the value (BLKID) at the top position of each block. All the contents of the four blocks (track information management file) are stored to for example a hard disk

by the recovery computer.

[0102]

The number of total tracks is obtained from data after the fourth byte of the track information management file. The 20-th byte of the track information area TRKINF-001, the value of the area CONNUM-001 of the first music program, and the value of the next area P-001 are obtained. The number of parts is obtained with the value of the area P-001. The values of the areas PRTSIZE of all parts of the track 1 of the area PRTINF is obtained. The number of total blocks (clusters) n is calculated and obtained.

[0103]

After the track information management file is obtained, the flow advances to step 102. At step 102, a voice data file (ATRAC3 data file) is searched. All blocks of other than the management file is searched from the flash memory. Blocks whose top value (BLKID) is A3D are collected.

[0104]

A block of which the value of the area CONNUM0 at the 16-th byte of A3Dnnnn is the same as that of the area CONNUM-001 of the first music program of the track information management file and of which the value of the area BLOCK SERIAL that starts from 20-th byte is 0 is searched. After the first block is obtained, a block (cluster) with the same value of the

area CONNUM value as the first block and of which the value of BLOCK SERIAL is incremented by 1 (1 = 0 + 1) is searched. After the second block is obtained, a block with the same value of the area CONNUM0 as the second block and of which the value of the area BLOCK SERIAL is incremented by 1 (2 = 1 + 1) is searched.

[0105]

By repeating the process, the ATRC3 data file is searched until n blocks (clusters) of the track 1 are obtained. When all the blocks (clusters) are obtained, they are successively stored to the hard disk.

[0106]

The same process for the track 1 is performed for the track 2. In other words, a block of which the value of the area CONNUM0 is the same as that of the area CONNUM-002 of the first music program of the track information management file and of which the value of the area BLOCK SERIAL that starts at the 20-th byte is searched. Thereafter, in the same manner as the track 1, the ATRAC3 data file is searched until the last block (cluster) n' is detected. After all blocks (clusters) are obtained, they are successively stored to the hard disk.

[0107]

By repeating the above-described process for all tracks (the number of tracks: m), all the ATRAC3

data is stored to the hard disk controlled by the recovering computer.

[0108]

At step 103, the memory card whose the FAT has been destroyed is re-initialized and then the FAT is reconstructed. A predetermined directory is formed in the memory card. Thereafter, the track information management file and the ATRAC3 data file for m tracks are copied from the hard disk to the memory card. Thus, the recovery process is finished.

[0109]

In the track information management file, an additonal information management file and data file, important parameters (in particular, codes in headers) may be recorded triply rather than doubly. When data is redundantly recorded, the same data may be recorded at any positions as long as they are apart from each other for one page or more.

[0110]

In the first and second embodiments, as an example of the recorder of the system audio set, a memory card recorder was described. According to the present invention, a digital signal reproduced by a CD player is stored to a hard disk. The hard disk is used as an audio server. The digital signal is moved from the hard disk to a memory card 40 having the above-described format. Thus, with the above-described

digital audio recorder or portable recorder, the user can listen to the reproduced digital audio data.

[0111]

Fig. 27 shows a storing apparatus having a hard disk drive such as a PC. Hereinafter the description, a storing apparatus is only referred as to the host or the host side. The hard disk drive is shown by 201 and a hard disk drive 201 is operated by the controller of CPU202. Thus, relating with CPU201, a non-volatile memory (the external NVRAM) 203, the operation button 204 and the display device 205 are attached.

[0112]

In addition, an ATRAC3 audio encoder/decoder 206 is disposed. An analog input signal 207 is supplied to an A/D converter 208. The A/D converter 208 converts the analog signal 207 into a digital audio signal. The audio encoder/decoder 206 compresses the digital audio signal that is output from the A/D converter 208 corresponding to ATRAC3. The host side decodes audio data stored in the hard disk drive 201. The audio encoder/decoder 206 decodes the audio data that is read from the hard disk drive 201 into a digital audio signal. The D/A converter 213 possibly outputs an analog audio signal 214.

[0113]

The compressed audio data is supplied from

the audio encoder/decoder 206 to a security block S-SAM (D) 212 of the host side. The security block S-SAM (D) 212 encrypts the compressed audio data. As with the audio recorder, the compressed audio data is encrypted using a contents key. The encrypted ATRAC3 data is stored to the hard disk drive 201 under the control of the CPU 202. In the case of the digital input, information such as ISRC (Industry Standard Recoding Code) and TOC (Table Of Content)_ID that identify music programs recorded on a disc can be obtained. The security block S-SAM (D) 212 generates a contents key and a contents cumulation number CONNUM for each contents title (audio file (track) in the first embodiment). In addition, each host is assigned a unique serial number. These values are stored in the hard disk drive 201 and/or the external non-volatile memory 203.

[0114]

To allow an encrypted ATRAC3 data file stored in the hard disk drive 201 to be reproduced by other than the unit (host) that has encrypted the ATRAC3 data file, the encrypted ATRAC3 data file is moved to the memory card 40. The moved data file is not left in the hard disk unlike with the copying process.

[0115]

Since the ATRAC3 data has been encrypted with a contents key, unless it is decrypted on the copied

side, it cannot be reproduced. However, when the contents key as an encrypting key is stolen, encrypted data can be easily decrypted. To prevent such a problem, the contents key itself is encrypted. The contents key is not exposed to the outside. For example, when ATRAC3 data is moved from the hard disk drive 201 to the memory card 40, the contents key is encrypted with a session key. The encrypted contents key is sent from the hard disk dive 201 to the memory card 40. The memory card 40 decrypts the contents key with the session key. Thereafter, the memory card 40 encrypts the contents data with a storage key thereof. The encrypted contents key is stored in the memory card 40.

[0116]

Likewise, when data is moved from the memory card 40 to the hard disk drive 201, the memory card 40 encrypts a contents key with a session key and sends the encrypted contents key to the hard disk drive 201. Thus, the value of the contents key stored in the hard disk drive 201 is different from the value of the contents key stored in the memory card 40. Thus, a pair of audio data and contents key should be stored on the moved side.

[0117]

Next, with reference to Fig. 28, the data move process will be described in detail. First of

all, a data move process for moving data formatted for the audio recorder shown in Fig. 1 and recorded in the memory card 40 to the hard disk drive 201 of the host side will be described. In the initial state of which the power of the host side is turned on, it is determined whether or not the memory card 40 has been attached. When the memory card 40 has been attached, the host side and the memory card 40 are authenticated each other. When they have been successfully authenticated, the host side and the memory card side share a session key Sek.

[0118]

Next, the host reads data from the memory card 40. According to the first embodiment of the present invention, the contents key CK is read from the reproduction track information management file PBLIST. In contrast, according to the second embodiment of the present invention, a contents key CK (DES (Data Encryption Standard) (Kstm, CK)) encrypted with a storage key Kstm that is unique to each memory card 40 is extracted from the track information area TRKINF. The DES (Kstm, CK) is sent from the host to the memory card 40. The memory card 40 decrypts the encrypted contents key DES (Kstm, CK) with the storage key Kstm. The decrypted contents key is encrypted with the session key Sek.

[0119]

The contents key DES (Sek, CK) encrypted with the session key Sek is sent from the memory card 40 to the host side. The host side decrypts the contents key CK with the session key Sek, re-encrypts the decrypted contents key CK with a storage key Kstd that is unique thereto, and stores the re-encrypted storage key to the hard disk drive 201. In other words, the key is stored as a new contents key. The storage keys Kstd and Kstm are stored in such a manner that their values cannot be read from the outside.

[0120]

In Fig. 28, a security block 212a of the host side and a security block of the memory card 40 authenticate each other and they share a session key Sek. The security block 212a supplies a storage key Kstd and a contents key CK to an encrypting device 212b. The encrypting device 212b creates an encrypted contents key DES (Kstd, CK).

[0121]

As denoted by a path 215, encrypted ATRAC3 data is moved from the memory card 40 to the host side. The ATRAC3 data is stored to the hard disk drive 201. In this case, as described with reference to Fig. 27, the track management information TRKINF recorded in the memory card 40 is sent to the host side along with a data file. In particular, the contents cumulation number (CONNUM), the S-SAM serial number, and the file

number FNM-nnnn for each music program are directly copied to the track information area TRKINF-nnnn and recorded as a track information area TRKINF of the host side. Unlike with the contents key, these attribute information is not encrypted.

[0122]

Unless these information is moved to the host side, even if audio data is stored to the hard disk drive 20, the audio data stored in the host cannot be decrypted. Unless the audio data stored in the hard disk is moved to the memory card, the audio data cannot be reproduced.

[0123]

The contents cumulation number CONNUM is a cumulation number of which each music program is recorded through encrypting devices of security blocks of the memory card 40 and the host side. The contents cumulation number CONNUM has a combination of 2^{32} = 4,200,000,000. The non-volatile memory of each encrypting device stores the last contents cumulation number. Thus, the contents cumulation number is not redundant in each memory card. The S-SAM serial number (SERIAL) is a number unique to each encrypting device. The S-SAM serial number has a combination of 2^{128}. Thus, the S-SAM serial number is not redundant. The file number FNM-nnnn is a number assigned to each ATRAC3 data file. The file number FNM-nnnn is assigned

by hardware. Thus, the file number FNM-nnnn may be redundant. Consequently, the contents cumulation number CONNUM and the S-SAM serial number (SERIAL) are added as auxiliary numbers. Thus, with a total of three types of numbers, a data file is recorded, the file can be uniquely existed.

[0124]

As described above, to perform an authenticating process and an encrypting process, the security block 212 of the host side creates or provides:

self unique number (S-SAM serial number),

contents key CK (created for each contents title),

storage key Kstd, and

session key Sek

[0125]

According to the second embodiment of the present invention, the hard disk drive 201 of the host side and/or the external non-volatile memory 203 has a track information area TRKINF paired with an audio data file. The track information area TRKINF contains:

file number FNM-nnnn,

encrypted contents key CK,

S-SAM serial number, and

contents cumulation number CONNUM.

[0126]

When digital data is directly recorded from for example the CD player 208 to the hard disk drive 201, the audio encoder/decoder 206 compresses audio data corresponding to ATRAC3. The security block 212 of the host side creates a contents key CK for each contents title (music program) and encrypts the contents key with the storage key Kstd unique thereto. The encrypting device 212c encrypts ATRAC3 data with the encrypted contents key DES (Kstd, CK) and stores the encrypted audio data 216 to the hard disk drive 201. At this point, the security block 212a of the host side creates the contents cumulation number CONNUM and the S-SAM (D) serial number for each music program. The contents cumulation number CONNUM and the S-SAM (D) serial number are stored as the track information area TRKINF to the hard disk drive 201. However, these attribute information is not encrypted with the storage key Kstd unlike with the contents key.

[0127]

In addition, the host itself decrypts and reproduces contents data stored in the hard disk drive 201. With the operation button portion 204, the user can record and reproduce contents data on the host side with reference to information displayed on the display device 205.

[0128]

When digital data is copied from the CD

player 209 to the hard disk drive 201 of the host side, the digital receiver 211 can obtain information that identifies a music program recorded on a CD (the information is for example TOC_ID or ISRC of each music program). When digital data received from the CD player 209 is copied, the digital receiver 211 assigns a directory name for each CD.

[0129]

In contrast, data can be moved from the host side to the memory card 40. In this case, the host side and the memory card 40 authenticate each other. When they have successfully authenticated each other, they share a session key Sek. The host reads a contents key DES (Kstd, CK) from the hard disk drive 201 and decrypts it with a storage key Kstd. The host encrypts the decrypted contents key with a session key Sek and sends the encrypted contents key DES (Sek, CK) to the memory card 40.

[0130]

The memory card 40 decrypts a contents key CK with a session key Sek. Thereafter, the memory card 40 re-encrypts the contents key CK with a storage key Kstm that is unique thereto. According to the first embodiment of the present invention, the encrypted contents key DES (Kstm, CK) is stored in the reproduction track information management file PBLIST and the ATRAC data file. According to the second

embodiment of the present invention, the encrypted contents key DES (Kstm, CK) is stored in the track information area TRKINF. Information (for example, contents cumulation number CONNUM and S-SAM () serial number) other than the contents key is not re-encrypted, but directly recorded.

[0131]

According to an embodiment of the present invention, to securely prevent audio data from being illegally copied, when the audio data is moved from the hose side to a memory card 40, information that represents a move history is stored to an external non-volatile memory 203. In other words, the hose side manages the move history that represents what music programs have been moved. Since the move history is stored to the external non-volatile memory 203 rather than the hard disk drive HDD 201, audio data recorded in the hard disk drive HDD 201 can be prevented from being illegally copied to a memory card. In other words, unless move information is recorded in the hard disk drive HDD 201, even if audio data recorded therein is illegally copied, moved data cannot be moved again.

[0132]

Fig. 29 shows a process for preventing audio data from being illegally copied. First of all, a copy process for copying audio data from a hard disk drive HDD 1 that stores audio data will be described. Before

performing a move process (that will be described later), 10 music programs stored in the hard disk drive HDD 1 are copied to a hard disk drive HDD 2. A host side CPU 202 and an external non-volatile memory 203 manage move history information. Thereafter, as described above, 10 music programs and a contents key that have been encrypted are moved from the hard disk drive HDD 1 to a first memory card 40X. In this case, as a precondition, the memory card 40X should have been correctly authenticated with the host side. When the audio data is moved, the encrypted contents key necessary for decrypting the audio data that has been moved to the memory card 40X is also sent to the memory card 40X. In such a manner, 10 music programs of audio data are completely moved from the hard disk drive HDD 1 to the memory card 40X.

[0133]

Next, a move process for moving 10 music programs of music data from the hard disk drive HDD 1 to the hard disk drive HDD 2 will be described. In this case, a second memory card 40Y is used. Since the host side has a security block 212, it correctly authenticates the memory card 40Y and shares a session key Sek with the memory card 40Y. Thus, the contents key CK encrypted with the session key Sek can be moved from the hard disk drive HDD 2 to the memory card 40Y. After the memory card 40Y is correctly authenticated,

when the encrypted data is moved to the memory card 40Y, the data stored therein can be decrypted and reproduced. When music programs are copied to a plurality of hard disk drives and the music programs are copied from a hard disk drive to a memory card, the music programs can be unlimitedly copied. Thus, the copyright of the music programs are violated. When 10 music programs stored in the hard disk drive HDD 1 on one host side are copied or moved to a hard disk drive on another host side, move history information stored in an external non-volatile memory NVRAM is prohibited from being copied/moved to the hard disk drive HDD 2.

[0134]

Corresponding to a flow chart shown in Fig. 30, the host side CPU 202 references the history information stored in the non-volatile memory 203 and determines whether or not to permit audio data to be moved. The memory card 40 sends a move request that designates a music program stored in the hard disk drive HDD 201 to the CPU 202 (at step S201). Thereafter, the CPU 202 checks the external non-volatile memory 203 for the move history of the designated music program (at step S202). In other words, the CPU 202 determines whether or not the designated music program has been moved corresponding to the move request (at step S203).

[0135]

When the determined result at step S203 is No, the flow advances to step S204. At step S204, the designated music program is moved from the host side hard disk drive HDD 201 to the memory card 40 (at step S204). In addition, the move history is recorded to the external non-volatile memory 203. When the determined result at step S203 is Yes, the host side CPU 202 prohibits the designated music program from being moved from the hard disk drive HDD 201 (at step S205). In this case, the display device 205 displays a message that represents that the designated music program has been moved. Alternatively, a synthesizing means may generates an audio message that represents that the designated music program has been moved.

[0136]

In the above description, data communication between a hard disk drive and a memory card that are storing units was described. Alternatively, a host having a hard disk drive (in this case, the host is for example a personal computer) may interface with a terminal unit of an electronic contents delivering system. In this case, a process similar to a move process performed between the hard disk drive and the memory card is performed between the terminal unit and the personal computer.

[0137]

In the above embodiment, the case that

contents data is audio data was described. Of course, the present invention can be applied to video data, program data, and so forth other than audio data. In addition, the present invention can be applied to other storage mediums such as a magneto-optical disc, a phase change type disc, and a semiconductor memory other than a hard disk.

[0138]

[Effect of the Invention]

According to the present invention, an encrypting device is also disposed on the storing unit side. A contents key encrypted with a session key and contents data (data file) encrypted with the contents key are received from a memory card as a storage medium. After the contents key is decrypted with the session key, the contents key is re-encrypted with a key unique to the storing unit. Since the contents key is re-keyed, even if the contents data is moved to other than the original memory card, the contents data can be decrypted. In addition, when contents data is moved from the storing unit to the memory card, the contents key is re-keyed. Thus, the contents moved to a memory card can be decrypted by another unit.

[0139]

In addition to a medium that stores contents data, move history information is stored in a non-volatile memory. Thus, contents data of a medium can

be securely prevented from being illegally copied to another medium.

[Brief Description of the Drawings]

[Fig. 1]

Block diagram showing the entire structure of the present invention.

[Fig. 2]

Block diagram showing the structure of a DSP according to the embodiment of the present invention.

[Fig. 3]

Block diagram showing the structure of a memory card according to the embodiment of the present invention.

[Fig. 4]

Schematic diagram showing the structure of the file system processes hierarchy of the flash memory according to the embodiment of the present invention.

[Fig. 5]

Schematic diagram showing the physical structure of the data in a flash memory according to the embodiment of the present invention.

[Fig. 6]

Schematic diagram showing a file convention according to the embodiment of the present invention.

[Fig. 7]

Schematic diagram showing the relation between files according to the embodiment of the

present invention.

[Fig. 8]

Schematic diagram showing the structure of a data file according to the present invention.

[Fig. 9]

Schematic diagram showing an example of the editing process of the data file according to the present invention.

[Fig. 10]

Schematic diagram showing the other example of the editing process of the data file according to the present invention.

[Fig. 11]

Schematic diagram showing the structure of the track information management file PBLIST according to the embodiment of the present invention.

[Fig. 12]

Schematic diagram showing a convention of the parts attribute information in the track information management file.

[Fig. 13]

Schematic diagram showing a convention of the parts attribute information in the track information management file.

[Fig. 14]

Schematic diagram showing the structure of the name file in the track information management file.

[Fig. 15]

Schematic diagram showing the structure of the name file in the track information management file.

[Fig. 16]

Schematic diagram showing the structure of the name file.

[Fig. 17]

Schematic diagram showing types of record mode and recording hours of each record mode according to the embodiment of the present invention.

[Fig. 18]

Schematic diagram showing the structure of the additional information management file according to the embodiment of the present invention.

[Fig. 19]

Schematic diagram showing the structure of the additional information data according to the embodiment of the present invention.

[Fig. 20]

Schematic diagram showing the example of the additional information according to the embodiment of the present invention.

[Fig. 21]

Schematic diagram showing the structure of the additional information according to the embodiment of the present invention.

[Fig. 22]

Schematic diagram explaining the example of the case that the additional information is the timestamp.

[Fig. 23]

Schematic diagram showing the example of the case that the additional information is the reproduction log file

[Fig. 24]

Schematic diagram showing the example of the case that the additional information is an artist name.

[Fig. 25]

Schematic diagram showing the example of the case that the additional information (an artist name) is deleted.

[Fig. 26]

Schematic diagram explaining the flow of the file recovery process.

[Fig. 27]

Block diagram explaining a move process according to the embodiment of the present invention.

[Fig. 28]

Schematic diagram explaining a re-keying operation in the move process.

[Fig. 29]

Schematic diagram explaining the prevention of the illegal copy by the copy of the hard disk.

[Fig. 30]

Flow chart explaining the prevention of the illegal copy by the copy of the hard disk.

[Description of Reference Numerals]

12 ... Audio decoder, 30 ... DSP, 40 ... Memory card, TRKLIST.MSF ... Track information management file, INFLIST.MSF ... Additional information management file, A3Dnnn.MSA ... Audio data file, 201 ... Hard disk drive, 202 ... CPU, 203 ... External non-volatile memory, 206 ... Audio encoder/decoder, 212 ... Security block, CK ... Contents key, Sek ... Session key, Kstm ... Unique storage key for the memory card, Kstd ... Unique storage key for storing apparatus

[Title of Document] Abstract

[Abstract]

[Subject]

The move of contents is performed smoothly with high security.

[Solving means]

The host has an attachable/detachable memory card 40 and the hard disk drive 201, shares the session key, when they have been successfully authenticated. contents key CK is encrypted by a unique storage key of the memory card that resides on the memory card 40. Contents key CK is decrypted that is encrypted by the session key Sek and transmitted to the host side. Encrypted data file and file attribute information are transmitted directly to the host side via a route 215. In the host side, contents key CK is decrypted by the session key Sek that is encrypted by a unique storage key Kstd of the host is allocated. Likewise contents are moved from the host to the memory card 40, the content's key is re-keyed. A move history information is managed on the non-volatile memory of the host side.

[Selected Drawing] Fig. 28

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT(S): Nobuyuki KIHARA and Teppei YOKOTA

SERIAL No.: 09/674,441 Group Art Unit: 2143

FILED: November 1, 2000 Examiner: Kyung H Shin

INVENTION: DATA PROCESSING APPARATUS, DATA PROCESSING METHOD, TERMINAL UNIT, AND TRANSMISSION METHOD OF DATA PROCESSING APPARATUS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

CERTIFIED TRANSLATION

Yuka NAKAMURA residing at c/o SUGIURA PATENT OFFICE, 7th floor, Ikebukuro Park Bldg., 49-7, Minami Ikebukuro 2-chome, Toshima-ku, Tokyo, JAPAN, declares:

(1) that she knows well both the Japanese and English languages;

(2) that she translated Japanese Application No. 11-055860 from Japanese to English;

(3) that the attached English translation is a true and correct translation of the above-identified Japanese Application to the best of her knowledge and belief; and

(4) that all statements made of her own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements are made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 USC 1001, and that such false statements may jeopardize the validity of the application or any patent issuing thereon.

November 17, 2004 Yuka Nakamura

Date Yuka NAKAMURA

[Title of Document] Specification

[Title of the Invention] Non-volatile Memory, Data Processing Apparatus, and Method

[Scope of Claims for a Patent]

[Claim 1]

A non-volatile memory from which a computer can read data and that is attachable to and detachable from the computer,

wherein the non-volatile memory stores:

data file,

first file management information for managing the data file, and

second file management information, used in a file management system of the computer, for managing the data file and the first file management information, and

wherein the first file management information has a fixed length whose data amount is any integer times the data amount of a predetermined data unit.

[Claim 2]

The non-volatile memory as set forth in claim 1,

wherein the first file management information has a header containing an identification code that represents the first file management information.

[Claim 3]

The non-volatile memory as set forth in claim 2,

wherein the identification code is repeatedly placed at positions apart from the header of the first file management information.

[Claim 4]

The non-volatile memory as set forth in claim 1,

wherein the data file managed corresponding to the first file management information contains a contents cumulation number and a sequence number at predetermined positions of each data unit.

[Claim 5]

The non-volatile memory as set forth in claim 1,

wherein the first file management information contains revision information at a predetermined position of each data unit, the revision information being varied whenever the data file is recorded.

[Claim 6]

The non-volatile memory as set forth in claim 1,

wherein the data file is divided into a plurality of parts, each of which has a fixed length, and

wherein the first file management information

contains management information for each data file and management information of each part of each data file.

[Claim 7]

The non-volatile memory as set forth in claim 1,

wherein the first file management information is redundantly recorded.

[Claim 8]

A data processing apparatus having an attachable/detachable non-volatile memory from which a computer can read data, the apparatus comprising:

a controlling portion for handling a data file, first file management information for managing the data file, and second file management information, used in a file management system, for managing the data file and the first file management information; and

a memory interface disposed between said controlling portion and the non-volatile memory,

wherein the data file, the first file management information, and the second file management information are stored from said controlling portion to the non-volatile memory through said memory interface,

wherein the data file, the first file management information, and the second file management information are read from the non-volatile memory to said

controlling portion through said memory interface, and

wherein the first file management information has a fixed length whose data amount is any integer times the data amount of a predetermined data unit of the non-volatile memory.

[Claim 9]

The data processing apparatus as set forth in claim 8,

wherein the first file management information has a header containing an identification code that represents the first file management information, and

wherein the first file management information is extracted from the non-volatile memory with reference to the identification code.

[Claim 10]

The data processing apparatus as set forth in claim 9,

wherein the identification code having the same value is repeatedly placed at positions apart from the header of the first file management information.

[Claim 11]

The data processing apparatus as set forth in claim 8,

wherein the data file managed corresponding to the first file management information contains a contents

cumulation number and a sequence number at predetermined positions of each data unit.

[Claim 12]

The data processing apparatus as set forth in claim 8,

wherein the first file management information contains revision information at a predetermined position of each data unit, the revision information being varied whenever the data file is recorded.

[Claim 13]

The data processing apparatus as set forth in claim 12,

wherein the first file management information contains the revision information at the beginning portion and the end portion of the first file management information,

wherein it is determined whether or not the revision information at the beginning portion of the first file management information matches the revision information at the end portion of the first file management information, and

wherein when the revision information at the beginning portion does not match the revision information at the end portion, it is determined that the first file management information has not been correctly rewritten.

[Claim 14]

The data processing apparatus as set forth in claim 8,

wherein the data file is divided into a plurality of parts, each of which has a fixed length, and

wherein the first file management information contains management information for each data file and management information of each part of each data file.

[Claim 15]

The data processing apparatus as set forth in claim 8,

wherein the first file management information is redundantly recorded.

[Claim 16]

A data processing method for a data processing apparatus from which a computer can read data and that is attachable to and detachable from the computer,

wherein the non-volatile memory stores:

a data file, first file management information for managing the data file, and

second file management information, used in a file management system, for managing the data file and the first file management information, and

wherein when the second file management information is destroyed, a file is restored with

reference to the first file management information.

[Claim 17]

The data processing method as set forth in claim 16, wherein the file is restored by the steps of:

searching and collecting the first file management information from the non-volatile memory;

searching and collecting the data file; and

initializing the non-volatile memory, re-structuring the second file management information, and storing the collected first file management information and the collected data file to the initialized non-volatile memory.

[Detailed Description of the Invention]

[0001]

[Technical Field to which the Invention belongs]

The present invention relates to a non-volatile memory, a data processing apparatus, and a method in which attachable/detachable memory card is used as a record media for audio data or so.

[0002]

[Prior Art]

EEPROM (Electrically Erasable Programmable ROM) that is an electrically rewritable memory requires a large space because each bit is composed of two transistors. Thus, the integration of EEPROM is

restricted. To solve this problem, a flash memory that allows one bit to be accomplished with one transistor using all-bit-erase system has been developed. The flash memory is being expected as a successor of conventional record mediums such as magnetic disks and optical discs.

[0003]

A memory card using a flash memory is also known. The memory card can be freely attached to an apparatus and detached therefrom. A digital audio recording/reproducing apparatus that uses a memory card instead of a conventional CD (Compact Disc: Trademark) or MD (Mini Disc: Trademark) can be accomplished.

[0004]

A file management system used for a conventional personal computer is named FAT (File Allocation Table). In the FAT system, when a particular file is defined, predetermined parameters are successively set to the file. Thus, the size of a file becomes variable. One file is composed of at least one management unit (sector, cluster, or the like). Data corresponding to the management unit is written to a table referred to as FAT. In the FAT file system, a file structure can be easily formed regardless of the physical characteristics of a record medium. Thus, the FAT file system can be used for a magneto-optical disc as well as

a floppy disk and a hard disk. In the above-mentioned memory card, the FAT file system is used.

[0005]

[Problem to be Solved by the Invention]

In the conventional FAT file system, once the FAT is destroyed, it cannot be almost recovered. Thus, as possible countermeasures, it is necessary to backup data to another medium. Among users of personal computers, such countermeasures are essential. Thus, the users should backup data as their responsibilities. However, it is troublesome for the users to backup data. In addition, to do that, another medium is required.

[0006]

Therefore, an object of the present invention is to provide a non-volatile memory, a data processing apparatus, and a method thereof that allow a file to be recovered even if a file management table is destroyed without need to make a backup file.

[0007]

[Means for Solving the Problem]

According to the invention disclosed in claim 1, a non-volatile memory from which a computer can read data and that is attachable to and detachable from the computer,

wherein the non-volatile memory stores:

data file,

first file management information for managing the data file, and

second file management information, used in a file management system of the computer, for managing the data file and the first file management information, and

wherein the first file management information has a fixed length whose data amount is any integer times the data amount of a predetermined data unit.

[0008]

According to the invention disclosed in claim 8, a data processing apparatus having an attachable/detachable non-volatile memory from which a computer can read data, the apparatus comprising:

a controlling portion for handling a data file, first file management information for managing the data file, and second file management information, used in a file management system, for managing the data file and the first file management information; and

a memory interface disposed between controlling portion and the non-volatile memory,

wherein the data file, the first file management information, and the second file management information are stored from controlling portion to the non-volatile memory through a memory interface,

wherein the data file, the first file management information, and the second file management information are read from the non-volatile memory to controlling portion through memory interface, and

wherein the first file management information has a fixed length whose data amount is any integer times the data amount of a predetermined data unit of the non-volatile memory.

[0009]

According to the invention disclosed in claim 16, a data processing method for a data processing apparatus from which a computer can read data and that is attachable to and detachable from the computer,

wherein the non-volatile memory stores:

a data file, first file management information for managing the data file, and

second file management information, used in a file management system, for managing the data file and the first file management information, and

wherein when the second file management information is destroyed, a file is restored with reference to the first file management information.

[0010]

[Embodiment of the Invention]

Next, an embodiment of the present invention

will be described. Fig. 1 is a block diagram showing the structure of a digital audio recorder (that has functions as a player and as a recorder) using a memory card according to an embodiment of the present invention. The digital audio recorder records and reproduces a digital audio signal using a attachable/detachable memory card. In reality, the recorder/player composes an audio system along with an amplifying unit, a speaker, a CD player, an MD recorder, a tuner, and so forth. However, it should be noted that the present invention can be applied to other audio recorders. For example, the present invention can be applied to a portable recording apparatus. In addition, the present invention can be applied to a recorder that records a digital audio data that is circulated as a satellite data communication, a digital broadcast, or Internet. Moreover, the present invention can be applied to a system that records/reproduces moving picture data and still picture data rather than audio data. The system according to the embodiment of the present invention can record and reproduce additional information such as picture and text other than a digital audio signal.

[0011]

The recorder has an audio encoder/decoder IC 10, a security IC 20, a DSP (Digital Signal Processor)

30. Each of these devices is composed of a one-chip IC. 40 is an attachable/detachable memory card to a recorder. The one-chip IC of the memory card 40 has flash memory (nonvolatile memory), a memory control block, and a security block. The security block has DES (Data Encryption Standard) encrypting circuit. According to the embodiment, the recording/reproducing apparatus may use a microcomputer instead of the DSP 30.

[0012]

The audio encoder/decoder IC 10 has an audio interface 11 and an encoder/decoder block 12. The encoder/decoder block 12 encodes a digital audio data corresponding to a highly efficient encoding method and writes the encoded data to the memory card 40. In addition, the encoder/decoder block 12 decodes encoded data that is read from the memory card 40. As the highly efficient encoding method, the ATRAC3 format that is a modification of the ATRAC (Adaptive Transform Acoustic Coding) format used in Mini-Disc is used.

[0013]

In the ATRAC3 format, audio data sampled at 44.1 kHz and quantized with 16 bits is processed. In the ATRAC3 format, the minimum data unit of audio data that is processed is a sound unit (SU). 1 SU is data of which data of 1024 samples (1024 x 16 bits x 2 channels) is

compressed to data of several hundred bytes. The duration of 1 SU is around 23 msec. In the ATRAC 3 , the data amount of audio data is compressed to data that is around 10 times smaller than that of original data. As with the Mini-Disc, the audio signal compressed and decompressed corresponding to the ATRAC3 format less deteriorates in the audio quality.

[0014]

A line input selector 13 selectively supplies the reproduction output signal of an MD, the output signal of a tuner, or a reproduction output signal of a tape to an A/D converter 14. The A/D converter 14 converts the selected line input signal to a digital audio signal (sampling frequency = 44.1 kHz; the number of quantizing bits = 16). A digital input selector 16 selectively supplies a digital output signal of an MD, a CD, or a CS (Satellite Digital Broadcast) to a digital input receiver 17. The digital input signal is transmitted through for example an optical cable. An output signal of the digital input receiver 17 is supplied to a sampling rate converter 15. The sampling rate converter 15 converts the digital input signal into a digital audio signal (sampling frequency = 44.1 kHz).

[0015]

The encoder/decoder block 12 of the audio

encoder/decoder IC 10 supplies encoded data to a DES encrypting circuit 22 through an interface 21 of the security IC 20. The DES encrypting circuit 22 has a FIFO 23. The DES encrypting circuit 22 is disposed so as to protect the copyright of contents. The memory card 40 also has a DES encrypting circuit. The DES encrypting circuit 22 of the recording/reproducing apparatus has a plurality of master keys and an apparatus-unique storage key. The DES encrypting circuit 22 also has a random number generating circuit. The DES encrypting circuit 22 can share an authenticating process and a session key with the memory card 40 that has the DES encrypting circuit. In addition, the DES encrypting circuit 22 can be repeatedly encrypted data with the storage key of the DES encrypting circuit.

[0016]

The encrypted audio data that is output from the DES encrypting circuit 22 is supplied to DSP (Digital Signal Processor) 30. The DSP 30 communicates with the memory card 40 through an interface. In this example, the memory card 40 is attached to an attaching/detaching mechanism (not shown) of the recording/reproducing apparatus. The DSP 30 writes the encrypted data to the flash memory of the memory card 40. The encrypted data is serially transmitted between the DSP 30 and the memory

card 40. In addition, an external SRAM (Static Random Access Memory) 31 is connected to the DSP 30. The SRAM 31 provides the recording/reproducing apparatus with a sufficient storage capacity so as to control the memory card 40.

[0017]

A bus interface 32 is connected to the DSP 30. Data is supplied from an external controller (not shown) to the DSP 30 through a bus 33. The external controller controls all operations of the audio system. The external controller supplies data such as a record command or a reproduction command that is generated corresponding to a user's operation through an operation portion to the DSP 30 through the bus interface 32. In addition, the external controller supplies additional information such as image information and character information to the DSP 30 through the bus interface 32. The bus 33 is a bidirectional communication path. Additional information that is read from the memory card 40 is supplied to the external controller through the DSP 30, the bus interface 32, and the bus 33. In reality, the external controller is disposed in for example an amplifying unit of the audio system. In addition, the external controller causes a display portion to display additional information, the operation

state of the recorder, and so forth. The display portion is shared by the audio system. Since data that is exchanged through the bus 33 is not copyright protected data, it is not encrypted.

[0018]

The encrypted audio data that is read from the memory card 40 by the DSP 30 is decrypted by the security IC 20. The audio encoder/decoder IC 10 decodes the encoded data corresponding to the ATRAC3 format. Output data of the audio encoder/decoder 10 is supplied to a D/A converter 18. The D/A converter 18 converts the output data of the audio encoder/decoder 10 into an analog signal. The analog audio signal is supplied to a line output terminal 19.

[0019]

The analog audio signal is supplied to an amplifying unit (not shown) through the line output terminal 19. The analog audio signal is reproduced from a speaker or a head set. The external controller supplies a muting signal to the D/A converter 18. When the muting signal represents a mute-on state, the external controller prohibits the audio signal from being output from the line output terminal 19.

[0020]

Fig. 2 is a block diagram showing the internal

structure of the DSP 30. Referring to Fig. 2, the DSP 30 comprises a core 34, a flash memory 35, an SRAM 36, a bus interface 37, a memory card interface 38, and inter-bus bridges. The DSP 30 has the same function as a microcomputer. The core 34 is equivalent to a CPU. The flash memory 35 stores a program that causes the DSP 30 to perform predetermined processes. The SRAM 36 and the external SRAM 31 are used as a RAM of the recording/reproducing apparatus.

[0021]

The DSP 30 controls a writing process for writing encrypted audio data and additional information to the memory card 40 corresponding to an operation signal such as a record command received through the bus interfaces 32 and 37 and a reading process for reading them therefrom. In other words, the DSP 30 is disposed between the application software side of the audio system that records/reproduces audio data and additional information and the memory card 40. The DSP 30 is operated when the memory card 40 is accessed. In addition, the DSP 30 is operated corresponding to software such as a file system.

[0022]

The DSP 30 manages files stored in the memory card 40 with the FAT system used in conventional personal

computers. In addition to the file system, according to the embodiment of the present invention, a management file is used. The management file will be descried later. The management file is used to manage data files stored in the memory card 40. The management file as the first file management information is used to manage audio data files. On the other hand, the FAT as the second file management information is used to mange all files including audio data files and management files stored in the flash memory of the memory card 40. The management file is stored in the memory card 40. The FAT is written to the flash memory along with the route directory and so forth before the memory card 40 is shipped. The details of the FAT will be described later.

[0023]

According to the embodiment of the present invention, to protect the copyright of data, audio data that has been compressed corresponding to the ATRAC3 format is encrypted. On the other hand, since it is not necessary to protect the copyright of the management file, it is not encrypted. There are two types of memory cards that are an encryption type and a non-encryption type. However, a memory card for use with the recorder that records copyright protected data is limited to the encryption type.

[0024]

Fig. 3 is a block diagram showing the internal structure of the memory card 40. The memory card 40 comprises a control block 41 and a flash memory 42 that are structured as a one-chip IC. A bidirectional serial interface is disposed between the DSP 30 of the recorder and the memory card 40. The bidirectional serial interface is composed of ten lines that are a clock line SCK for transmitting a clock signal that is transmitted along with data, a status line SBS for transmitting a signal that represents a status, a data line DIO for transmitting data, an interrupt line INT, two GND lines, two INT lines, and two reserved lines.

[0025]

The clock line SCK is used for transmitting a clock signal in synchronization with data. The status line SBS is used for transmitting a signal that represents the status of the memory card 40. The data line DIO is used for inputting and outputting a command and encrypted audio data. The interrupt line INT is used for transmitting an interrupt signal that causes the memory card 40 to interrupt the DSP 30 of the recorder. When the memory card 40 is attached to the recorder, the memory card 40 generates the interrupt signal. However, according to the embodiment of the present invention,

since the interrupt signal is transmitted through the data line DIO, the interrupt line INT is grounded.

[0026]

A serial/parallel converting, parallel/serial converting, and interface block (S/P, P/S, I/F block) 43 is an interface disposed between the DSP 30 of the recorder and the control block 41 of the memory card 40. The S/P, P/S, and IF block 43 converts serial data received from the DSP 30 of the recorder into parallel data and supplies the parallel data to the control block 41. In addition, the S/P, P/S, and IF block 43 converts parallel data received from the control block 41 into serial data and supplies the serial data to the DSP 30. When the S/P, P/S, and IF block 43 receives a command and data through the data line DIO, the S/P, P/S, and IF block 43 separates them into these that are normally accessed to the flash memory 42 and those that are encrypted.

[0027]

In the format of which data is transmitted through the data line DIO, after a command is transmitted, data is transmitted. The S/P, P/S, and IF block 43 detects the code of a command and determines whether the command and data are those that are normally accessed or those that are encoded. Corresponding to the

determined result, the S/P, P/S, and IF block 43 stores a command that is normally accessed to a command register 44 and stores data that is normally accessed to a page buffer 45 and a write register 46. In association with the write register 46, the memory card 40 has an error correction code encoding circuit 47. The error correction code encoding circuit 47 generates a redundant code that is an error correction code for data temporarily stored in the page buffer 45.

[0028]

Output data of the command register 44, the page buffer 45, the write register 46, and the error correction code encoding circuit 47 is supplied to a flash memory interface and sequencer (hereinafter, referred to as memory I/F and sequencer) 51. The memory IF and sequencer 51 is an interface disposed between the control block 41 and the flash memory 42 and controls data exchanged therebetween. Data is written to the flash memory through the memory IF and sequencer 51.

[0029]

Audio data that has been compressed corresponding to the ATRAC3 format and written to the flash memory (hereinafter, this audio data is referred to as ATRAC3 data) is encrypted by the security IC 20 of the recorder/player and the security block 52 of the memory

card 40 so as to protect the copyright of the ATRAC3 data. The security block 52 comprises a buffer memory 53, a DES encrypting circuit 54, and a nonvolatile memory 55.

[0030]

The security block 52 of the memory card 40 has a plurality of authentication keys and a unique storage key for each memory card. The nonvolatile memory 55 stores a key necessary for encrypting data. The key stored in the nonvolatile memory 55 cannot be analyzed. According to the embodiment, for example, a storage key is stored in the nonvolatile memory 55. The security block 52 also has a random number generating circuit. The security block 52 authenticates an applicable recorder/player and shares a session key therewith. In addition, the security block 52 re-encrypts contents with the storage key through the DSE encrypting circuit 54.

[0031]

For example, when the memory card 40 is attached to the recorder, they are mutually authenticated. The security IC 20 of the recorder and the security block 52 of the memory card 40 mutually authenticate. When the recorder has authenticated the attached memory card 40 as an applicable memory card and the memory card 40 has authenticated the recorder as an

applicable recorder, they are mutually authenticated. After the mutual authenticating process has been successfully performed, the recorder/player and the memory card 40 generate respective session keys and share them with each other. Whenever the recorder/player and the memory card 40 authenticate each other, they generate respective session keys.

[0032]

When contents are written to the memory card 40, the recorder/player encrypts a contents key with a session key and supplies the encrypted data to the memory card 40. The memory card 40 decrypts the contents key with the session key, re-encrypts the contents key with a storage key, and supplies the contents key to the recorder. The storage key is a unique key for each memory card 40. When the recorder/player receives the encrypted contents key, the recorder/player performs a formatting process for the encrypted contents key, and writes the encrypted contents key and the encrypted contents to the memory card 40.

[0033]

Data that is read from the flash memory 42 is supplied to the page buffer 45, the read register 48, and the error correction circuit 49 through the memory IF and the sequencer 51. The error correcting circuit 49

corrects an error of the data stored in the page buffer 45. Output data of the page buffer 45 that has been error-corrected and the output data of the read register 48 are supplied to the S/P, P/S, and IF block 43. The output data of the S/P, P/S, and IF block 43 is supplied to the DSP 30 of the recorder through the above-described serial interface.

[0034]

When data is read from the memory card 40, the contents key encrypted with the storage key and the contents encrypted with the block key are read from the flash memory 42. The security block 52 decrypts the contents key with the storage key. The security block 52 re-encrypts the decrypted content key with the session key and transmits the re-encrypted contents key to the recorder. The recorder decrypts the contents key with the received session key and generates a block key with the decrypted contents key. The recorder successively decrypts the encrypted ATRAC3 data.

[0035]

A configuration ROM 50 is a memory that stores partition information, various types of attribute information, and so forth of the memory card 40. The memory card 40 also has an erase protection switch 60. When the switch 60 is in the erase protection position,

even if a command that causes the memory card 40 to erase data stored in the flash memory 42 is supplied from the recorder side to the memory card 40, the memory card 40 is prohibited from erasing the data stored in the flash memory 42. An OSC cont. 61 is an oscillator that generates a clock signal that is the reference of the timing of the process of the memory card 40.

[0036]

Fig. 4 is a schematic diagram showing the hierarchy of the processes of the file system of the computer system that uses a memory card as a storage medium. On the hierarchy, the top hierarchical level is an application process layer. The application process layer is followed by a file management process layer, a logical address management layer, a physical address management layer, and a flash memory access layer. In the above-mentioned hierarchical structure, the file management process layer is the FAT file system. Physical addresses are assigned to individual blocks of the flash memory. The relation between the blocks of the flash memory and the physical addresses thereof does not vary. Logical addresses are addresses that are logically handled on the file management process layer.

[0037]

Fig. 5 is a schematic diagram showing the

physical structure of data handled in the flash memory 42 of the memory card 40. In the memory 42, a data unit (referred to as segment) is divided into a predetermined number of blocks (fixed length). One block is divided into a predetermined number of pages (fixed length). In the flash memory, data is erased as each block at a time. Data is written to the flash memory 42 or read therefrom as a page at a time. The size of each block is the same. Likewise, the size of each page is the same. One block is composed of page 0 to page m. For example, one block has a storage capacity of for example 8 KB (kilobytes) or 16 KB. One page has a storage capacity of 512 B (bytes). When one block has a storage capacity of 8 KB, the total storage capacity of the flash memory 42 is 4 MB (512 blocks) or 8 MB (1024 blocks). When one block has a storage capacity of 16 KB, the total storage capacity of the flash memory 42 is 16 MB (1024 blocks), 32 MB (2048 blocks), or 64 MB (4096 blocks).

[0038]

One page is composed of a data portion of 512 bytes and a redundant portion of 16 bytes. The first three bytes of the redundant portion is an overwrite portion that is rewritten whenever data is updated. The first three bytes successively contain a block status area, a page status area, and an update status area. The

remaining 13 bytes of the redundant portion are fixed data that depends on the contents of the data portion. The 13 bytes contain a management flag area (1 byte), a logical address area (2 bytes), a format reserve area (5 bytes), a dispersion information ECC area (2 bytes), and a data ECC area (3 bytes). The dispersion information ECC area contains redundant data for an error correction process against the management flag area, the logical address area, and the format reserve area. The data ECC area contains redundant data for an error correction process against 512-byte data.

[0039]

The management flag area contains a system flag (1: user block, 0: boot block), a conversion table flag (1: invalid, 0: table block), a copy prohibition flag (1: OK, 0: NG), and an access permission flag (1: free, 0: read protect).

[0040]

The first two blocks - blocks 0 and 1 are boot blocks. The block 1 is a backup of the block 0. The boot blocks are top blocks that are valid in the memory card. When the memory card is attached to the recorder, the boot blocks are accessed at first. The remaining blocks are user blocks. Page 0 of the boot block contains a header area, a system entry area, and a boot

and attribute information area. Page 1 of the boot block contains a prohibited block data area. Page 2 of the boot block contains a CIS (Card Information Structure)/IDI (identify Drive Information) area.

[0041]

The header area of the boot block contains a boot block ID and the number of effective entries. The system entries are the start position of prohibited block data, the data size thereof, the data type thereof, the data start position of the CIS/IDI area, the data size thereof, and the data type thereof. The boot and attribute information contains the memory card type (read only type, rewritable type, or hybrid type), the block size, the number of blocks, the number of total blocks, the security/non-security type, the card fabrication data (date of fabrication), and so forth.

[0042]

Since the flash memory has a restriction for the number of rewrite times due to the deterioration of the insulation film, it is necessary to prevent the same storage area (block) from being concentratedly accessed. Thus, when data at a particular logical address stored at a particular physical address is rewritten, updated data of a particular block is written to a non-used block rather than the original block. Thus, after data is

updated, the relation between the logical address and the physical address changes. This process is referred to as swap process. Consequently, the same block is prevented from being concentratedly accessed. Thus, the service life of the flash memory can be prolonged.

[0043]

The logical address associates with data written to the block. Even if the block of the original data is different from the block of updated data, the address on the FAT does not change. Thus, the same data can be properly accessed. However, since the swap process is performed, a conversion table that correlates logical addresses and physical addresses is required (this table is referred to as logical-physical address conversion table). With reference to the logical-physical address conversion table, a physical address corresponding to a logical address designated on the FAT is obtained. Thus, a block designated with a physical address can be accessed.

[0044]

The DSP 30 stores the logical-physical address conversion table in the SRAM. When the storage capacity of the RAM is small, the logical-physical address conversion table can be stored to the flash memory. The logical-physical address conversion table correlates

logical addresses (2 bytes) sorted in the ascending order with physical addresses (2 bytes). Since the maximum storage capacity of the flash memory is 128 MB (8192 blocks), 8192 addresses can be assigned with two bytes. The logical-physical address conversion table is managed for each segment. Thus, the size of the logical-physical address conversion table is proportional to the storage capacity of the flash memory. When the storage capacity of the flash memory is 8 MB (two segments), two pages are used as the logical-physical address conversion table for each of the segments. When the conversion table is stored in the flash memory, a predetermined one bit of the management flag area in the redundant portion in each page represents whether or not the current block is a block containing the logical-physical address conversion table.

[0045]

The above-described memory card can be used with the FAT file system of a personal computer system as with the disc shaped record medium. The flash memory has an IPL area, a FAT area, and a route directory area (not shown in Fig. 5). The IPL area contains the address of a program to be initially loaded to the memory of the recorder. In addition, the IPL area contains various types of memory information. The FAT area contains

information with respect to blocks (clusters). The FAT has defined unused blocks, next block number, defective blocks, and last block number. The route directory area contains directory entries that are a file attribute, an update date [day, month, year], file size, and so forth.

[0046]

According to the embodiment of the present invention, in addition to the file management system defined in the format of the memory card 40, the management file is used for managing tracks and parts of music files. The management file is recorded to a user block of the flash memory 42 of the memory card 40. Thus, as will be described later, even if the FAT of the memory card 40 is destroyed, a file can be recovered.

[0047]

The management file is generated by the DSP 30. When the power of the recorder is turned on, the DSP 30 determines whether or not the memory card 40 has been attached to the recorder. When the memory card has been attached, the DSP 30 authenticates the memory card 40. When the DSP 30 has successfully authenticated the memory card 40, the DSP 30 reads the boot block of the flash memory 42. Thus, the DSP 30 reads the physical-logical address conversion table and stores the read data to the SRAM. The FAT and the route directory have been written

to the flash memory of the memory card 40 before the memory card 40 is shipped. When data is recorded to the memory card 40, the management file is generated.

[0048]

In other words, a record command issued by the remote controller of the user or the like is supplied to the DSP 30 from the external controller through the bus and the bus interface 32. The encoder/decoder IC 10 compresses the received audio data and supplies the resultant ATRAC3 data to the security IC 20. The security IC 20 encrypts the ATRAC3 data. The encrypted ATRAC3 data is recorded to the flash memory 42 of the memory card 40. Thereafter, the FAT and the management file are updated. Whenever a file is updated (in reality, whenever the recording process of audio data is completed), the FAT and the management file stored in the SRAMs 31 and 36 are rewritten. When the memory card 40 is detached or the power of the recorder is turned off, the FAT and the management file that are finally supplied from the SRAMs 31 and 36 are recorded to the flash memory 42. Alternatively, whenever the recording process of audio data is completed, the FAT and the management file written in the flash memory 42 may be rewritten. When audio data is edited, the contents of the management file are updated.

[0049]

In the data structure according to the embodiment, additional information is contained in the management file. The additional information is updated and recorded to the flash memory 42. In another data structure of the management file, an additional information management file is generated besides the track management file. The additional information is supplied from the external controller to the DSP 30 through the bus and the bus interface 32. The additional information is recorded to the flash memory 42 of the memory card 40. Since the additional information is not supplied to the security IC 20, it is not encrypted. When the memory card 40 is detached from the recorder or the power thereof is turned off, the additional information is written from the SRAM of the DSP 30 to the flash memory 42.

[0050]

Fig. 6 is a schematic diagram showing the file structure of the memory card 40. As the file structure, there are a still picture directory, a moving picture directory, a voice directory, a control directory, and a music (HIFI) directory. According to the embodiment, music programs are recorded and reproduced. Next, the music directory will be described. The music directory

has track information management file TRKLIST.MSF, the backup file TRKLISTB.MSF that is a backup file of TRKLIST.MSF, INFLIST. MSF that describes artist name, ISRC code, time stamp and still picture data, and 3Dnnnn. MSF. That is a ATRAC 3 data file. TRKLIST.MSF has NAME1 and NAME2. NAME1 has memory card name, and these program name block(for 1 byte code) in which program name data is described with the letter code of ASCII/8859-1. NAME2 has memory card name, and these music name block(for 2 byte code) in which program name data is described with the letter code of MS-JIS/Hankul/Chinese or the like.

[0051]

Fig. 7 shows the relation between the track information management file TRKLIST, the areas NAME1 and NAME2, and the ATRAC3 data file A3Dnnnn. The file TRKLIST is a fixed-length file of 64k bytes (= 16 k x 4). An area of 32k bytes of the file is used for managing tracks. The remaining area of 32k bytes is used to contain the areas NAME1 and NAME2. Although the areas NAME1 and NAME2 for program names may be provided as a different file as the track information management file, in a system having a small storage capacity, it is convenient to totally manage the track information management file and program name files.

[0052]

The track information area TRKINF-nnnn and part information area PRTINF-nnnn of the track information management file TRKLIST are used to manage the data file A3Dnnnn and the additional information INFLIST.MSN. Only the ATRAC3 data file A3Dnnnn is encrypted. In Fig. 6, the data length in the horizontal direction is 16 bytes (0 to F). A hexadecimal number in the vertical direction represents the value at the beginning of the current line.

[0053]

Next, with reference to Fig. 8, the relation between music programs and ATRAC3 data files will be described. One track is equivalent to one music program. In addition, one music program is composed of one ATRAC3 data (see Fig. 8). The ATRAC3 data file is audio data that has been compressed corresponding to the ATRAC3 format. The ATRAC3 data file is recorded as a cluster at a time to the memory card 40. One cluster has a capacity of 16 KB. Some pluralities of files are not contained in one cluster. The minimum data erase unit of the flash memory 42 is one block. In the case of the memory card 40 for music data, a block is a synonym of a cluster. In addition, one cluster is equivalent to one sector.

[0054]

One music program is basically composed of one part. However, when a music program is edited, one music program may be composed of a plurality of parts. The connection of parts of a music program is described in track information management file TRKLIST.MSN. A part is a unit of data that is successively recorded. Normally, one track is composed of one part. The maximum number of parts is limited. The relationship between the number of parts P and the number of tracks T is (P= 2043-4xT) When 1 track is composed by 2039 parts, for example, no part can be assigned to the second part, and the second file cannot be generated.

[0055]

SU is the minimum unit of a part. In addition, SU is the minimum data unit in the case that audio data is compressed corresponding to the ATRAC3 format. 1 SU is audio data of which data of 1024 samples at 44.1 kHz (1024 x 16 bits x 2 channels) is compressed to data that is around 10 times smaller than that of original data. The duration of 1 SU is around 23 msec. Normally, one part is composed of several thousand SU.

[0056]

Fig. 8 is a schematic diagram showing the file structure in the case that two music programs of a CD or the like are successively recorded. The first program

(file 1) is composed of for example five clusters. Since one cluster cannot contain two files of the first program and the second program, the file 2 starts from the beginning of the next cluster. Thus, the end of the part 1 corresponding to the file 1 is in the middle of one cluster and the remaining area of the cluster contains no data. Likewise, the second music program (file 2) is composed of one part.

[0057]

There are four types of edit processes that are a divide process, a combine process, an erase process, and a move process. The divide process is performed to divide one track into two portions. When the divide process is performed, the number of total tracks increases by one. In the divide process, one file is divided into two files on the file system. Thus, in this case, the reproduction management file and the FAT are updated. The combine process is performed to combine two tracks into one track. When the combine process is performed, the number of total tracks decreases by one. In the combine process, two files are combined into one file on the file system. Thus, when the combine process is performed, the reproduction management file and the FAT are updated. The erase process is performed to erase a track. The track numbers after the track that has been

erased decrease one by one. The move process is performed to change the track sequence. Thus, the reproduction management file and the FAT are updated. The other operation of the move process is to translate a track from memory card to other media, such as a hard disk. Comparing the copy process in which a replica of a track is generated, the move process means only translation of track position. Thus, a replica is not generated in performing the move process.

[0058]

Fig. 9 is a schematic diagram showing the combined result of two programs (file 1 and file 2) shown in Fig. 8. As a result of the combine process, the combined file is composed of two parts. Fig. 10 is a schematic diagram showing the divided result of which one program (file 1) is divided in the middle of the cluster 2. By the divide process, the file 1 is composed of clusters 0, 1, and the beginning portion of cluster 2. The file 2 is composed of the end portion of cluster 2 and clusters 3 and 4.

[0059]

When edit process described above is performed, only file management information TRKLIST.MSN corresponding to a block (cluster) that includes edited point is rewritten. Since, to rewrite ATTRAC3 data file

corresponding to performed edit process needs much time. Considering the matters described above, the concept "part" is introduced.

[0060]

Fig. 11 shows the detailed structure of the track information management file TRKLIST.MSN. In the track information management file TRKLIST, one cluster (block) is composed of 16k bytes. The size and data of the file TRKLISTB.MSN are the same as those of the backup file TRKLISTB that follows the file TRKLISTB.MSN. The first 32 bytes of the track information management file are used as a header. The unit divided per 8 bytes from the beginning of file is called slot. However, the unit divided per 16 bytes is called slot concerning in the track information management file. The head that is located in the first slot of file includes data described below in order.

[0061]

BLK ID-TL0/TL1 (4 bytes)

Fixed value(TL0=0x544C2D30, TL1=0x544C2D31)

T-TRK (2 bytes)

Represents about the number of total tracks

Mcode (2 bytes)

Represents code to identify the maker and model of the recorder/player.

Mcode is management code to identify the maker of recorder that records some information to memory card. Mcode is given when licenced. The type code is managed by makers individually that are licenced .

REVISION (4 bytes)

Represents the number of TRKLIST rewrite times, REVISION is incremented in terms of recording.

YMDhms (4 bytes)

Represents update date and time data.

N1 (OP)(1 byte)

Represents the sequential number of the memory card (numerator side). When one memory card is used, the value of the area N1 is 0x01. "OP" means optional term.

N2 (OP)(1 byte)

Represents the sequential number of the memory card (denominator side). When one memory card is used, the value of the area N2 is 0x01.

MSID (OP)(2 bytes)

Represents the ID of a memory card. When a plurality of memory cards is used, the value of the area MSID of each memory card is the same (T.B.D.). (T.B.D. (to be defined) represents that this value may be defined in future).

S-TRK (2 bytes).

Represents a special track (T.B.D.). Normally,

the value of the area S-TRK is 0x0000.

PASS (OP)(2 bytes)

Represents a password (T.B.D.).

APP (OP)(2 bytes)

Represents the definition of a reproduction application (T.B.D.) (normally, the value of the area APP is 0x0000).

INF-S (OP)(2 bytes)

Represents the additional information pointer of the entire memory card. When there is no additional information, the value of the area INF-S is 0x00.

S_YTMDhms (OP)(4 bytes)

Represents TRKLIST.MSF.'s update date and time.

The last 16 bytes of the file TRKLIST are used for an area BLKID-TL0, an area Mcode, and an area REVISION that are the same as those of the header. The backup file TRKLISTB contains the above-described header. In this case, the header contains an area BLKID-TL1, an area Mcode, and an area REVISION.

[0062]

While data is being recorded to a memory card, it may be mistakenly or accidentally detached or the power of the recorder/player may be turned off. When such an improper operation is performed, a defect should be detected. As described above, the REVISION area is

placed at the beginning and end of each block. Whenever data is rewritten, the value of the REVISION area is incremented. If a defect termination takes place in the middle of a block, the value of the REVISION area at the beginning of the block does not match the value of the REVISION area at the end of the block. Thus, such a defect termination can be detected. Since there are two REVISION areas, the abnormal termination can be detected with a high probability. When an abnormal termination is detected, an alarm such as an error message is generated.

[0063]

In addition, since the fixed value BLKID-TL0 is written at the beginning of one block (16 KB), when the FAT is destroyed, the fixed value is used as a reference for recovering data. In other words, with reference to the fixed value, the type of the file can be determined. Since the fixed value BLKID-TL0 is redundantly written at the header and the end portion of each block, the reliability can be secured. Alternatively, the same reproduction management file can be redundantly recorded.

[0064]

The data amount of an ATRAC3 data file is much larger than that of the track information management file. For example, several thousand blocks may be included in an ATRAC3 data file. In addition, as will be

described later, a block number BLOCK SERIAL is added to ATRAC3 data file. However, since a plurality of ATRAC3 files are recorded to the memory card, to prevent them from become redundant, both CONNUM0 and BLOCK SERIAL are used. Otherwise, when the FAT is destroyed, it will be difficult to recover the file.

[0065]

Likewise, the maker code (Mcode) is redundantly recorded at the beginning and the end of each block so as to identify the maker and the model in such a case that a file has been improperly recorded in the state that the FAT has not been destroyed.

[0066]

The header is followed by a track information area TRKINF.NSN that describes information of each track (a music program), and a part information area PRTINF that describes information of each track(music programs). Fig.11 shows theses areas preceded by the area TRKLIST. The lower portion of the area TRKLISTB shows the detailed structure of these areas. In Fig. 11, a hatched area represents an unused area. On the track information area TRKINF-nnn, the information described below is recorded.

[0067]

T0 (1 byte)

Fixed value (T0 = 0x74)

LT (1 byte)

Represents Reproduction restriction is regulated or not.

INF-nnn (OP) (2 bytes)

Represents the additional information pointer (0 to 409) of each track. 00: music program without additional information.

FNM-nnn (4 bytes)

Represents the file number (0x0000 to 0xFFFF) of an ATRAC3 data file.

The number nnnn (in ASCII) of the ATRAC3 data file name (A3Dnnnn) is converted into 0xnnnnn.

CONTENTS KEY (8 bytes)

Represents special value for each music program. The value of CONTENTS KEY is encrypted in the security block of the memory card and then stored.

S-SAM(D) SERIAL (16 bytes)

Represents serial number of apparatus that has recorded the memory card.

APP_CTL (4 bytes) (OP)

Represents an application parameter (T.B.D.) (Normally, the value of the area APP_CTL is 0x0000).

CONNUM (4 bytes)

Represents a unique value cumulated for each music program. The value is stored in the security block

of the recorder, so as not to overlap in a memory card.

P-nnn (2 bytes)

Represents the number of parts (1 to 2039) that compose a music program.

XT (OP)(2 bytes)

Represents the reproduction duration (SU) from the point designated by INX.

0000: no setting; FFFF: up to end of music program

INX-nnn(OP) (2 bytes)

Represents a pointer that indicates specified part of music program. INX-nnn shows the number of relative SU.

YMDhms-S (4 bytes)

Represents reproduction end date and time of track with reproduction restriction.

When not used, the value of the YMDhms-S is 0x00000000.

YMDhms-E (4 bytes)

Represents reproduction end date and time of track with reproduction restriction.

When not used, the value of the YMDhms-E is 0x00000000.

MT (1 byte)

Represents the maximum number of permitted

reproduction times.

When not used, the value of MT is 0x00.

CT (1 byte)

Represents number of reproduction of the track to which reproduction condition is regulated.

When not used, the value of the area CT is 0x00.

CC (1 byte)

Controls the copy operation. 00: copy prohibited, 01: one time copy operation permitted, 10: unlimited copy operation permitted. When one time copy operation permitted, copied track is prohibited to be copied.

CN (1 byte)

CN is concerned about number of permitted copy times. 00: Copy prohibited, 01 to 0xFE: Number of times, 0xFF: Unlimited copy times, it is valid about one time copy. Copied time is counted in terms of copy operation is done.

In the part information area PRINTF-nnn, the part information is described as illustrated below.

[0068]

PR (1 byte)

Fixed value (PR = 0 x 50).

A-nnnn (2 byte)

Represents the attribute information on parts. A-nnnn consists of mode(1 byte) and SCMS(Serial Copy Management System) information(1 byte).

PRTSIZE-nnnn (4 bytes)

Represents the cluster size of a part(2 bytes), start SU(1 byte), and end SU(1 byte).

PRTKEY-nnnn (8 bytes)

PRTKEY-nnnn is used with CONTENTSKEY to generate block key to encrypt music data.

PRTKEY-nnnn's initial value = 0, and PRTKEY-nnnn is incremented +1 whenever a part generates by progression of edit operation.

The mode information that indicates the mode of ATRAC3, and that is described by lower byte of A-nnnn information, is defined as illustrated in Fig. 12. Fig. 12 shows the number of bytes and recording period(in case 64MB), data translation rate, and compression rate about 6 kinds of mode, such as HQ, SP, CD, LP1, LP2, mono. Fig. 13 shows the content of information described by upper byte. SCMS information is formed by combining Bit 5 and Bit 6, as shown in Fig. 13.

[0069]

Fig. 14 shows the detailed structure of the area NAME1 (for one byte code area). Each of the areas NAME1 and NAME2 (that will be described later) is

segmented with eight bytes. Thus, their one slot is composed of eight bytes. At 0x8000 that is the beginning of each of these areas, a header is placed. The header is followed by a pointer and a name. The last slot of the area NAME1 contains the same areas as the header.

[0070]

BLK ID-NM1 (4 bytes)

Represents the contents of a block (fixed value) (NM1 = 0x4E4D2D31).

Mcode(2 bytes)

Represents code to identify the maker and model of recorder/player.

PNM1-nnn (OP)(4 bytes)

Represents the pointer to the area NM1 (for one byte code).

PNM1-S represents the pointer to a name representing a memory card.

nnn (= 1 to 408) represents the pointer to a music program title.

The pointer represents the start position (2 bytes) of the block, the character code type (2 bits), and the data size (14 bits).

The starting point is represented by byte offset value(0x000-ox3989) from the beginning of NM1 region.

The types of letter code are (0:ASCII, 1:ASCII + kana letter, 2:revised 8859-1)

Data size(14 bits) is the value(0x000-0x398C) that is calculated by adding letter data and terminal(0x00).

NM1-nnn (OP)

Represents the memory card name and music program title for one byte code (variable length). An end code (0x00) is written at the end of the area.

Fig. 15 shows the detailed data structure of the area NAME2 (for two byte code). At 0x8000 that is the beginning of the area, a header is placed. The header is followed by a pointer and a name. The last slot of the area NAME2 contains the same data as the header.

[0071]

BLK ID-NM2 (4 bytes)

Represents the contents of a block (fixed value) (NM2 = 0x4E4D2D32).

Mcode (2 bytes)

Represents the code to identify the maker and model of the recorder/player.

PNM2-nnn(OP) (4 bytes)

Represents the pointer to NM2(2 bytes code)

PNM2-S is the pointer to the name representing

the memory card.

nnn (= 1 to 408) is the pointer to a music program title.

The pointer represents the start position (2 bytes) of the block, the character code type (2 bits), and the data size (14 bits).

The starting point is represented by byte offset value(0x000-ox3989) from the beginning of NM2 region.

The types of letter code are (0:Japanese(MS-JIVES), 1:Korean(KC C5601-1989), 2:Chinese(BG2312-80))

Data size(14 bits) is the value(0x000-0x398C) that is calculated by adding letter data and terminal(0x00).

NM2-nnn (OP)

Represents the memory card name and music program title for two byte code (variable). An end code (0x0000) is written at the end of the area.

Fig. 16 shows the data arrangement (for one block) of the ATRAC3 data file A3Dnnnn in the case that 1 SU is composed of N bytes. In this file, one slot is composed of eight bytes. Fig. 15 shows the values of the top portion (0x0000 to 0x3FF8) of each slot. The first four slots of the file are used for a header. BLOCK SEED is redundantly recorded in the slot that is located

before the last slot of block. BLK ID-A3D and Mcode is recorded in the last slot.

[0072]

BLK ID-A3D (4 bytes)

fixed value(A3D=0x41324420) to designate the content of block.

Mcode (2 bytes)

Represents the code to identify the maker and model of the recorder/player.

Mcode is needed to rewritten when edit process is performed.

BLOCK SEED (8 bytes)

BLOCK SEED is used to generate block key that is necessary for encrypting process.

The beginning value of BLOCK SEED is put by calculation of random number performed by security block, and the value corresponding to the following block is incremented +1.

The same value is written at the starting and the ending of the block, as a countermeasure against error.

BLOCK SEED is not needed to rewritten even when edit process is performed.

CONNUM (4 bytes)

CONNUM is the initial contents cumulation

number.

Initially the value of CONNUM is the same value as the value of CONNUM of TRKLIST.MSF.

CONNUM is not needed to rewritten even when edit process is performed.

BLOCK SERIAL (4 bytes)

BLOCK SERIAL is put to 0 to the beginning block, and incremented +1 to following blocks.

BLOCK SERIAL is not needed to rewritten even when edit process is performed.

INITIALIZATION VECTOR (8 bytes)

INITIALIZATION VECTOR is the initial value that is necessary for encrypting/decrypting process on ATTRAC3 data.

The value of INITIALIZATION VECTOR corresponding to the beginning of contents is 0.

The value of INITIALIZATION VECTOR corresponding to the following block is the last encrypted value in the encrypted values corresponding to the last SU.

INITIALIZATION VECTOR is not needed to rewritten even when edit process is performed.

The header is followed by the sound unit data SU-nnnn. SU is the data formed by compressing 1024 samples. The data amount of SU is varied in terms of

mode. SU is not needed to rewritten even when edit process is performed. Fig. 17 shows the relationship between mode and the data amount of SU, the number of SU per 1 block, and the amount of residual (reserved) data, transferring rate, and time.

[0073]

As one example, a case in which 64MB memory card is used, and CD mode is operated will be referred. 64MB memory card has 3968 blocks, and 1 SU corresponds to 320 bytes in CD mode. Thus, 1 Block has 51 SU's. Therefore, 1block corresponds (1024/44100) x 51 x (3968-16) = 4680(sec) = 78(min). Transferring rate is (44100/1024) x 320 x 8 = 110250 bps

[0074]

Fig. 18 shows the detailed data structure of the additional information management file INFLIST.MSN that contains additional information.
INFLIST.MSN is a part of track information management file TRKLIST.MSF. Thus INFLIST.MSN is fragmented in terms of 16 bytes, starting from the beginning of file. The following header is placed at the beginning (0x0000) of the file INFLIST. The header is followed by the following pointer and areas.

[0075]

BLK ID-INF (4 bytes)

Represents the contents of the block (fixed value) (INF = 0x494E464F).

T-DAT (2 bytes)

Represents the number of total data areas (0 to 409).

Mcode (2 bytes)

Represents the code to identify the maker and model of the recorder/player

YMDhms (4 bytes)

Represents the record updated date and time.

INF-nnnn (4 bytes)

INF-nnnn is the pointer to the area DATA of the additional information (variable length, as 2 bytes (slot) at a time).

The start position is represented with the high order 16 bits (0000 to FFFF).

INF-nnnn Represents the offset value from the beginning of DataSlot-0000 (0x0800).

The data size is represented with low order 16 bits (0001 to 7FFF). A disable flag is set at the most significant bit. MSB = 0 (Enable), MSB = 1 (Disable)

The data size represents the total data amount of the music program.

(The data starts from the beginning of each slot. The non-data area of the slot is filled with 00.)

The first INF represents a pointer to additional information of the entire album (normally, INF-409).

Fig. 19 shows the structure of additional information. An 8-byte header described below is placed at the beginning of one additional information data area.

[0076]

IN (1 byte)

fixed value (IN=0x69)

ID (1 byte)

Represents the larger classification of additional information.

ID is called key ID compared to sub ID.

SID(1 byte)

Represents the classification of sub ID(T. B. D.)

SIZE (2 bytes)

Represents the size of additional information for each ID in terms of slot (1 to 7FFF).

Disable flag is set on most significant bit MSB. MSB=0(Enable), MSB=1(Disable)

Mcode (2 bytes)

Represents the code to identify the maker and model of the recorder that has recorded.

Fig. 20 shows an example of additional information. When size is 0x8xxx, erase or disable data is indicated. Each additional information is to be identified with the code in header, such as key ID and SID. However, the values of key ID and SID are not shown, since the values of key ID and SID are not defined. The additional information includes copy right code ISRC(International Standard Recording Code), the information about music such as the name of Composer and artist, and hardware management information. The letter code that is described in the starting 2 byte of data added to music information.

[0077]

Fig. 21 shows the structure of one additional information. In reference to the structure shown in Fig. 21, some concrete example of additional information will be described. Fig. 22 shows an example in which additional information is time stamp. The time stamp is time stamp on recording process as shown in Fig. 20. The data is YMDhms, and 00 is written in residual area. Fig. 23 shows an example in which additional information is reproduction log file. Year-month-day(YMD) data and hour-minute-second(HMS) data is written.

[0078]

Fig. 24 shows an example in which additional

information is artist name + ISRC code + TOCID. In this example, an artist name is described in 1 byte code. 00 is written in residual area of the slot. ISRC code is described as a data in the following slot. TOCID data is described as a data in the following slot. When the additional data shown in Fig 24 is erased, the additional data shown in Fig 24 is rewritten to the data shown in Fig 25. In other word, SIZE is to be (8xxx).

[0079]

According to the embodiment of the present invention, in addition to the file system defined as a format of the memory card, the track information management file TRKLIST.MSF for music data is used. Thus, even if the FAT is destroyed, the file can be recovered. Fig. 26 shows a flow of a file recovering process. To recover the file, a computer that operates with a file recovery program and that can access the memory card and a storing device (hard disk, RAM, or the like) connected to the computer are used. The computer has a function equivalent to the DSP30. The operation that is performed in the first step 101 will be described.

[0080]

All blocks of the flash memory whose FAT has been destroyed are searched for TL-0 as the value (BLKID)

at the top position of each block. In addition, all the blocks are searched for NM-1 as the value (BLKID) at the top position of each block. Thereafter, all the blocks are searched for NM-2 as the value (BLKID) at the top position of each block. All the contents of the four blocks (track information management file) are stored to for example a hard disk by the recovery computer.

[0081]

The number of total tracks is obtained from data after the fourth byte of the track information management file. The 20th byte of the track information area TRKINF-001, the value of the area CONNUM-001 of the first music program, and the value of the next area P-001 are obtained. The number of parts is obtained with the value of the area P-001. The values of the areas PRTSIZE of all parts of the track 1 of the area PRTINF is obtained. The number of total blocks (clusters) n is calculated and obtained.

[0082]

After the track information management file is obtained, the flow advances to step 102. At step 102, a voice data file (ATRAC3 data file) is searched. All blocks of other than the management file is searched from the flash memory. Blocks whose top value (BLKID) is A3D are collected.

[0083]

A block of which the value of the area CONNUM0 at the 16-th byte of A3Dnnnn is the same as that of the area CONNUM-001 of the first music program of the track information management file and of which the value of the area BLOCK SERIAL that starts from 20-th byte is 0 is searched. After the first block is obtained, a block (cluster) with the same value of the area CONNUM value as the first block and of which the value of BLOCK SERIAL is incremented by 1 (1 = 0 + 1) is searched. After the second block is obtained, a block with the same value of the area CONNUM0 as the second block and of which the value of the area BLOCK SERIAL is incremented by 1 (2 = 1 + 1) is searched.

[0084]

By repeating the process, the ATRC3 data file is searched until n blocks (clusters) of the track 1 are obtained. When all the blocks (clusters) are obtained, they are successively stored to the hard disk.

[0085]

The same process for the track 1 is performed for the track 2. In other words, a block of which the value of the area CONNUM0 is the same as that of the area CONNUM-002 of the first music program of the track information management file and of which the value of the

area BLOCK SERIAL that starts at the 20-th byte is searched. Thereafter, in the same manner as the track 1, the ATRAC3 data file is searched until the last block (cluster) n' is detected. After all blocks (clusters) are obtained, they are successively stored to the hard disk.

[0086]

By repeating the above-described process for all tracks (the number of tracks:m), all the ATRAC3 data is stored to the hard disk controlled by the recovering computer.

[0087]

At step 103, the memory card whose the FAT has been destroyed is re-initialized and then the FAT is reconstructed. A predetermined directory is formed in the memory card. Thereafter, the track information management file and the ATRAC3 data file for m tracks are copied from the hard disk to the memory card. Thus, the recovery process is finished.

[0088]

In the management file and data file, important parameters (in particular, codes in headers) may be recorded triply rather than doubly. When data is redundantly recorded, the same data may be recorded at any positions as long as they are apart from each other

for one page or more.

[0089]

[Effects of the Invention]

According to the present invention, in addition to the file management information defined in the nonvolatile memory such as the FAT, another (second) file management information is generated and stored in the nonvolatile memory. In the second file management information, a fixed length identification code that represents file management information is added. Thus, even if the FAT is destroyed, a file can be easily recovered with the file management information. According to the present invention, since the file management information has a fixed length identification value, the efficiency of the recovering process can be improved. Thus, it is not necessary for the user to make a backup file.

[0090]

Moreover, in the file management information, important parameters are redundantly recorded. Thus, important parameters can be securely protected. In addition, since information that represents the number of rewrite times of a file is recorded at a position apart from the other by 1 page unit or more, a trouble in the middle of the rewriting process of a file can be

detected. Moreover, when a trouble takes place, the cause of the trouble can be easily obtained.

[0091]

According to the present invention, in addition to the concept of the file, part management information is stored. Even if one track (music program) is composed of a plurality of parts, they can be easily managed. Moreover, since the part management information for parts that compose a track is handled along with the track management information (TRKINF) for tracks, the process can be more easily performed than that of Mini-Disc using links (Link-P).

[Brief Description of the Drawings]

[Fig. 1]

Block diagram showing the structure of the embodiment of the present invention.

[Fig. 2]

Block diagram showing the internal structure of a DSP in the embodiment of the present invention.

[Fig. 3]

Block diagram showing the internal structure of a memory card in the embodiment of the present invention.

[Fig. 4]

Schematic diagram showing a data structure of a file system operating layer in the flash memory in the

embodiment of the present invention.

[Fig. 5]

Schematic diagram showing the physical structure of data in a flash memory in the embodiment of the present invention.

[Fig. 6]

Schematic diagram showing the regulation of file in the embodiment of the present invention.

[Fig. 7]

Schematic diagram showing a relationship between files in the embodiment of the present invention.

[Fig. 8]

Schematic diagram showing the data structure of a data file in the embodiment of the present invention.

[Fig. 9]

Schematic diagram showing one example of the edit process in the embodiment of the present invention.

[Fig. 10]

Schematic diagram showing another example of the edit process in the embodiment of the present invention.

[Fig. 11]

Schematic diagram showing the data structure of track information management file.

[Fig. 12]

Schematic diagram showing the regulation to part attribute information of track information management file.

[Fig. 13]

Schematic diagram showing the regulation to part attribute information in track information management file.

[Fig. 14]

Schematic diagram showing the data structure of name file in track information management file.

[Fig. 15]

Schematic diagram showing the data structure of name file in track information management file.

[Fig. 16]

Schematic diagram showing the data structure of data file.

[Fig. 17]

Schematic diagram showing a various kinds of record mode and record period in various record mode.

[Fig. 18]

Schematic diagram showing the data structure of additional information management file.

[Fig. 19]

Schematic diagram showing the data structure of additional information management file.

[Fig. 20]

Schematic diagram showing a example of the data structure of additional data in the embodiment of the present invention.

[Fig. 21]

Schematic diagram showing the data structure of the additional information data in the embodiment of the present invention.

[Fig. 22]

Schematic diagram showing the data structure in the case that additional information data is a time stamp.

[Fig. 23]

Schematic diagram showing the data structure in the case that additional information data is a reduction log.

[Fig. 24]

Schematic diagram showing the data structure in the case that additional information data is an artist name.

[Fig. 25]

Schematic diagram showing the data structure in the case that additional information data (an artist name) is erased.

[Fig. 26]

Schematic diagram showing the process to recover the file.

[Description of Reference Numerals]

10 ... Audio encoder/decoder, 20 ... Security ID, 30 ... DSP, 40 ... Memory card, 42 ... Flash memory, 52 ... Security block, TRKLIST.MSF ... Track information Management file, INFLIST.MSF ... Additional information management file, A3Dnnn.MSA ... Audio data file

[Title of Document] Abstract

[Abstract]

[Subject]

In a case of writing or reading an audio and other data in the flash memory of attachable/detachable memory card, the file can be recovered whenever the file management system FAT is destroyed.

[Solving means]

Flash memory is attachable/detachable to the recorder as a memory card. The reproduction management file, FAT, and audio data file are recorded to the flash memory. The reproduction management file is composed of 64KB. and parameters for managing track are dually recorded in 32KB. BLKID for detecting the kind of block, REVISION for indicating the number of rewriting and so forth are described on the header. Important parameters in headers are dually recorded on the end of 1 block. The track information of TRKINF. and part information of PRTINF. are described by each truck unit as the management information.

[Selected Drawing] Fig. 11